Bio-inspired Optimization Algorithms for Smart Antennas

Virgilio Zúñiga Grajeda

Thesis submitted for the degree of Doctor of Philosophy
The University of Edinburgh

June, 2011

Declaration of originality

I hereby declare that the research recorded in this thesis and the thesis itself was composed and originated entirely by myself in the School of Engineering at the University of Edinburgh.

Virgilio Zúñiga Grajeda, June 2011.

Acknowledgements

I would like to deeply thank my supervisor Prof. Tughrul Arslan. His guidance and helpfulness were paramount in producing successful research. I would also like to thank Dr. Ahmet Erdogan who was always available when I needed his advice. I extend a sincere thanks to all members of System Level Integration Group at Edinburgh University for their support and motivation. This research has been supported by the Mexican National Council for Science and Technology (CONACyT). Studentship 181512.

Publications

- **V. Zuniga**, N. Haridas, A. T. Erdogan, and T. Arslan, "Effect of a central antenna element on the directivity, half-power beamwidth and sidelobe level of circular antenna arrays," *NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009)*, pp. 252–256, San Francisco, California, USA, July 29 - August 1 2009.

- **V. Zuniga**, A. T. Erdogan, and T. Arslan, "Adaptive radiation pattern optimization for antenna arrays by phase perturbations using particle swarm optimization," *NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2010)*, pp. 209–214, Anaheim, California, USA, June 15-18 2010.

- **V. Zuniga**, A. T. Erdogan, and T. Arslan, "Control of adaptive rectangular antenna arrays using particle swarm optimization," *Loughborough Antennas & Propagation Conference (LAPC-2010)*, pp. 385–388, Loughborough UK, November 8-9 2010.

- N. H. Noordin, **V. Zuniga**, A. O. El-Rayis, N. Haridas, A. T. Erdogan, and T. Arslan, "Uniform circular arrays for phased array antenna," *Loughborough Antennas & Propagation Conference (LAPC-2011)*, Loughborough UK, November 14-15 2011. To be published.

Abstract

This thesis studies the effectiveness of bio-inspired optimization algorithms in controlling adaptive antenna arrays. Smart antennas are able to automatically extract the desired signal from interferer signals and external noise. The angular pattern depends on the number of antenna elements, their geometrical arrangement, and their relative amplitude and phases. In the present work different antenna geometries are tested and compared when their array weights are optimized by different techniques. First, the Genetic Algorithm and Particle Swarm Optimization algorithms are used to find the best set of phases between antenna elements to obtain a desired antenna pattern. This pattern must meet several restraints, for example: Maximizing the power of the main lobe at a desired direction while keeping nulls towards interferers. A series of experiments show that the PSO achieves better and more consistent radiation patterns than the GA in terms of the total area of the antenna pattern. A second set of experiments use the Signal-to-Interference-plus-Noise-Ratio as the fitness function of optimization algorithms to find the array weights that configure a rectangular array. The results suggest an advantage in performance by reducing the number of iterations taken by the PSO, thus lowering the computational cost. During the development of this thesis, it was found that the initial states and particular parameters of the optimization algorithms affected their overall outcome. The third part of this work deals with the meta-optimization of these parameters to achieve the best results independently from particular initial parameters. Four algorithms were studied: Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing and Hill Climb. It was found that the meta-optimization algorithms Local Unimodal Sampling and Pattern Search performed better to set the initial parameters and obtain the best performance of the bio-inspired methods studied.

Contents

Chapter 1

Introduction

1.1 Motivation

In the last few years, the use of mobile and wireless communication devices like mobile phones, global positioning systems and personal digital assistants has increased in such a way that the network bandwidth is affected. One way to tackle this problem is to design antenna architectures that meet the requirements of communication systems. In the recent years, antenna designers have benefited from the use of simulation software tools that allow the exploration of a large variety of configurations before fabrication. A large variety of antennas have been developed to date [1, 2, 3]; they range from simple structures such as monopoles and dipoles to complex structures such as phased arrays. One way to improve the capabilities of an antenna is to consider a set of individual elements in a geometrical configuration creating an antenna array; the overall radiation pattern of the array is obtained with the summation of each radiated field of every individual element. The interaction amongst all radiation patterns depends on the geometry of the array (number of elements, distance between elements, etc.), where the pattern of the elements should interfere constructively in the direction of the signals-of-interest (SOI) and destructively in any other direction or signals-not-of-interest (SNOI). Amongst others, one of the characteristics that determines the shape of the radiation pattern is the geometric configuration of the array (linear, rectangular, circular, etc.). Numerous studies of antenna array geometries have

been conducted in the past [4, 5], where, due to their symmetry, uniform circular arrays (UCA) were found to have advantages over other geometries in terms of scanning abilities. These advantages motivated the work presented in this thesis in Chapter 4 where a Uniform Circular Array is considered and its geometric structure is modified. The circular configuration is rearranged in such a way that one of the elements is placed in the centre of the array. As shown by the results in [6], compared to the original UCA, this modification allows better values of directivity and half-power beamwidth to be obtained.

Another way to solve the problems faced by wireless communication systems is to employ smart antennas. Smart antennas are systems that combine multiple antenna elements in order to automatically optimize their radiation pattern in response to the signal environment. Smart antennas are able to extract the desired signal from interferer signals and external noise. This is achieved by radiating power toward a particular direction and excluding undesired signals from other incidence angles. Although the concept of smart antennas has been around since 1950, the technology required to implement them has only emerged in the last few years. The development of digital signal processing permits smart antennas to execute operations digitally which were once done by analog hardware. As mentioned previously, the radiation pattern of an antenna array can be controlled by changing the characteristics of the system, for example the relative amplitude and phases of the array elements depend on the angular pattern that must be achieved. By changing the relative phases of array elements, a process called steering, an array is capable of focusing its main beam towards a particular direction. This manipulation of the phases of each element is achieved by signal processing; thus, an algorithm running in a computer control or intelligence calculating these phases is needed. Due to the recent development of modern computers, the application of numerical optimization techniques to antenna design has become possible. Evolutionary optimization algorithms have been applied to adapting the response of an antenna array in order to reject interference. Genetic algorithms have been used to tune the amplitude and phase of adaptive antenna arrays in order to place nulls in the directions of undesired signals [7, 8]. Other bio-inspired algorithms like the Bees Algorithm have also been used for the pattern synthesis

of a linear antenna array with prescribed nulls [9]. Moreover, algorithms like the Particle Swarm Optimization (PSO) [10] also have been found to be effective. In [11], PSO was used to reconfigure phase-differentiated array antennas by finding element excitations that produce a main beam with low sidelobes. These developments in the bio-inspired arena motivated the research done in this thesis in Chapter 5, where the use of the Particle Swarm Optimization technique to find the optimal radiation pattern of an adaptive antenna is investigated.

During the development of this thesis, it was found that the performance of an optimization algorithm to solve a given problem depends heavily on its initial parameters. To enhance the effectiveness of the algorithm, these parameters should be carefully selected according to the problem to be solved. For example, the GA has crossover and mutation rates which will affect the overall ability of the algorithm to converge to the desired solution. By modifying these parameters, a good balance between exploration and exploitation can be achieved. Traditionally, the behavioural parameters have been chosen according to numerous experiments done by researchers. Parameters can also be selected according to mathematical analysis as shown in [12], in which the PSO algorithm is analysed and graphical parameter selection guidelines are provided. The selection of parameters can be divided in two cases: parameter tuning and parameter control [13]. In parameter control the parameter values change during the optimization run. An initial parameter value is needed and it has to suit the control strategies which can be deterministic, adaptive, or self-adaptive [14]. On the other hand, in parameter tuning the values do not change during the run but there are still a large number of combinations depending on the number of parameters (variables). In the last chapter of this thesis, the parameters of algorithms Differential Evolution, Simulated Annealing, Hill Climb and Particle Swarm Optimization are selected using a technique called meta-optimization. This process consists of using another optimization algorithm to find good behavioural parameters. Meta-optimization allows for an objective way to find the most suitable set of parameters for a given optimization method and problem to be solved. Different antenna problems like maximizing the signal-to-interference-plus-noise ratio, are solved using meta-optimized parameters. Moreover, antenna synthesis

problems proposed in the literature [15], namely the optimization of distances between antenna elements, are tackled as well using meta-optimization techniques to enhance the efficiency and efficacy of optimization algorithms.

1.2 Research Goals

The aim of this thesis is to investigate different approaches for using bio-inspired algorithms to enhance the capabilities of smart antennas. This is achieved by proposing strategies to improve the effectiveness of these algorithms when tackling adaptive antenna array problems. The main objectives of this work can be summarized in the following points:

1. To understand the geometrical characteristics of antenna arrays and how they determine the shape of their radiation pattern. This is carried out by analysing different antenna configurations, particularly uniform circular arrays. To carry out measurements of directivity, half-power beamwidth and sidelobe levels to provide a better understanding of the effect that displacement of the antenna elements have on the overall performance. To study the effect that these geometrical changes have on the range of frequencies at which the antenna can transmit.

2. To study the feasibility of using bio-inspired algorithms like Particle Swarm Optimization and Genetic Algorithms among others to obtain the optimal antenna radiation pattern for a given problem. One possible requirement being the steering of the main beam towards a certain direction while keeping low power levels in the direction of interferers. In particular, to examine the process of digitally shifting the phase weights of an adaptive antenna array. To investigate different approaches of computing the appropriate fitness function in order to obtain the best performance of each algorithm. To carry out a comparison of the different algorithmic approaches suggested in the past and draw conclusions about their performance. Specifically in terms of the number of fitness function evaluations which are of paramount importance in real-time systems like mobile devices.

3. To investigate and implement different strategies to enhance the effectiveness of optimization algorithms. To apply these strategies to a number of optimization algorithms in order to compare their efficiency and efficacy. To carry out a statistical analysis of the data obtained.

4. To provide a better understanding of the impact initial parameters have on the ability of optimization algorithms to find the best solution to adaptive antenna array problems. To generate and provide data that can be used in the development and implementation of smart antenna systems.

1.3 Thesis Overview

This thesis is organized as follows:

- Chapter 2 presents a general overview of smart antennas and their importance in wireless communication systems. The fundamentals of antenna arrays are also described including the definition of figures like directivity, half-power beamwidth, sidelobe level, etc. The basic antenna array geometry configurations are also explained. Linear, rectangular and circular antenna arrays are discussed and their mathematical description is presented. Additionally, a break down of the formula for obtaining the signal-to-interference-plus-noise ratio is provided.

- Chapter 3 describes the optimization algorithms used throughout this thesis. These algorithms are: Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) which will be used in Chapters 5 and 6. Differential Evolution (DE), Simulated Annealing (SA), Hill Climb (HC), Pattern Search (PS) and Local Unimodal Sampling (LUS) are studied in Chapter 7 for meta-optimization.

- In Chapter 4, an analysis of the effect of a central antenna element on the radiation pattern in a uniform circular antenna arrays is presented. A modification of the array geometry is considered in which one of the antenna elements is placed in the centre of the array. The corresponding

array factor is adjusted to describe the geometric configuration that includes the central antenna element. Array configurations with different numbers of antenna elements are tested and the results on directivity and half-power beamwidth are presented. Using simulation software, a 6-element circular antenna array was designed and the directivity for a range of frequencies was obtained. Moreover, additional results were obtained for a range of transmission frequencies and for different phases of the central element.

- Chapter 5 shows the process of obtaining an optimal radiation pattern for a linear antenna array using the Particle Swarm Optimization algorithm. In order to control the main beam and to steer it towards a desired direction while keeping null at interferers a set of phase shift weights is generated. The fitness function that allows the calculations of the phase shift weights is presented and a comparison between the standard genetic algorithm and the particle swarm optimization is provided.

- Chapter 6 describes the use of the Particle Swarm Optimization algorithm to generate a set of array weights for a uniform planar rectangular array. The aim is to maximize the power towards a desired direction while minimizing it in the direction of interferers. A fitness function based on the Signal-to-Interference-plus-Noise Ratio is employed. The results are compared with those obtained by the Genetic Algorithm.

- In Chapter 7, the initial parameters of the algorithms Differential Evolution, Simulated Annealing, Hill Climb and Particle Swarm Optimization are selected using a technique called meta-optimization. A group of algorithms, namely Pattern Search, Local Unimodal Sampling as well as DE and PSO are selected to act as a second layer of optimization over the mentioned techniques. Meta-landscapes, as well as statistics, are obtained for each meta-optimization experiment. A similar antenna problem to that considered in the previous chapters is solved using the obtained meta-optimized parameters.

- Chapter 8 presents a summary and the conclusions of this thesis including the main contributions of this research.

Chapter 2

Literature Review: Antennas

This chapter provides a general overview of smart antennas and their role in mobile communication systems. The fundamentals of antenna arrays are described as they will be used throughout this thesis. Figures-of-merit like directivity, half-power beamwidth, sidelobe level etc. are explained along with the basic antenna array configurations. The geometry of antenna arrays: linear, rectangular and circular are discussed as well as their mathematical description. In the final section, the signal-to-interference-plus-noise ratio is described.

2.1 Smart Antennas

A smart antenna system combines multiple antenna elements to optimize its radiation and/or reception pattern automatically in response to the signal environment. Adaptive antennas are able to automatically extract the desired signal from interferer signals and external noise. This is achieved by radiation power towards a particular direction and excluding undesired signals from other incidence angles [16]. This concept is shown in Figure 2.1.

Although the technology required to execute large number of calculations is new, the concept of smart antennas emerged in the late 50s [17]. The use of multiple antennas together with a complex signal processing unit has been applied in defence systems [16]. These systems had a high cost which prevented them from being used commercially. It was only in recent years that new technolo-

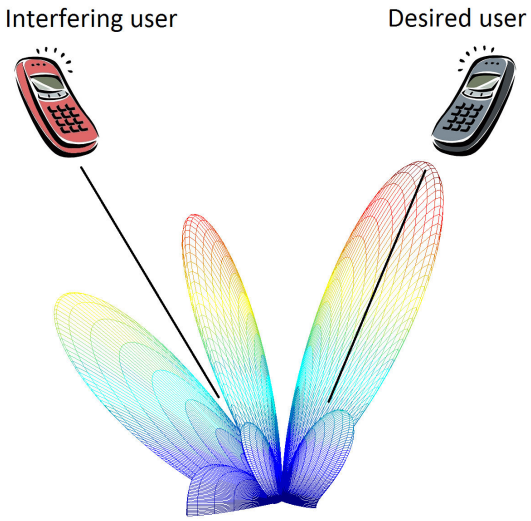


Figure 2.1: Desired and interfering signals.

gies like digital signal processing (DSP) permitted adaptive arrays to perform digitally where this was once implemented in analog hardware [18]. This can be achieved by using digital-to-analog converters connected to antenna elements controlled by a range of voltages [19]. DSP can be implemented using field programmable gate arrays (FPGA) allowing parallel processing which would make the signal processing run faster. Software-based algorithms have also made smart antennas practical for wireless communications. The global demand for cellular communications systems and wireless sensor networks justifies the development of intelligent antennas so as to increase the coverage area, maintain a high quality of service and eliminate interference with other users.

The goal of a smart antenna system is to augment the signal quality through a more focused transmission of its radio signal, thus providing higher system capacities. This allows higher signal-to-interference ratios, lower power levels, and permits greater frequency reuse. This concept is called space division multiple access (SDMA) [16]. Another benefit of smart antennas is spatial diversity. Information from the array is used to minimize the effective delay spread of the channel allowing higher data rates by nulling multipath signals. Higher data rates reduce fading in the received signal and suppress co-channel interference. Multipath reduction not only benefits wireless communications but also applies

to applications of radar systems. These are only a few benefits of smart antennas. In short, the following list enumerates some of their advantages [20]:

- Reduction of sidelobe levels or null steering
- Increased frequency reuse
- Blind adaptation
- Improved direction-of-arrival (DOA) estimation
- Improved array resolution
- Multiple-input-multiple-output (MIMO) compatibility
- Tracking of moving sources
- Increased degrees of freedom

Smart antenna patterns are controlled via algorithms based on certain criteria. This criteria could be maximizing the signal-to-interference-plus-noise ratio (SINR), minimizing the variance, minimizing the *mean-square error* (MSE), steering towards a desired signal, nulling interfering signals etc. When using adaptive algorithms, the digital beamforming process is referred to as *adaptive beamforming*. A diagram of typical adaptive antenna array is shown in Figure 2.2. The array consists in a set of antenna elements connected to a receiver through amplitude and phase shift weights. By using an adaptive algorithm, the antenna is capable of adjust itself to a changing signal environment.

2.2 Figures-of-merit in Array Theory

A typical antenna pattern is shown in Figure 2.3 as a polar plot in linear units. The *main lobe* (or main beam) is the lobe containing the direction of maximum radiation. There are also usually a series of lobes smaller than the main lobe. Any lobe other than the main lobe is called a *minor lobe*. A *side lobe* is defined as a radiation lobe in any direction other than that of the intended lobe [20].

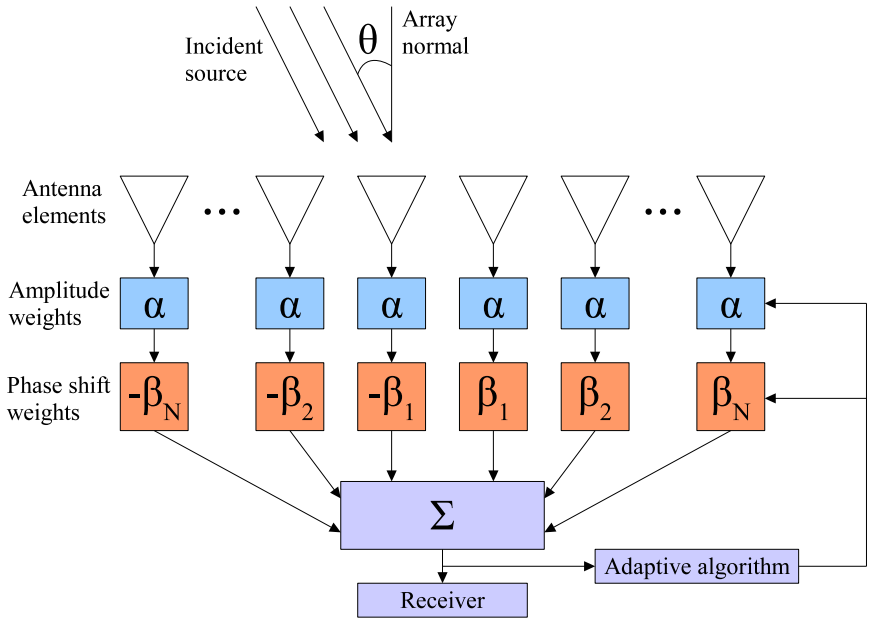


Figure 2.2: Adaptive linear array.

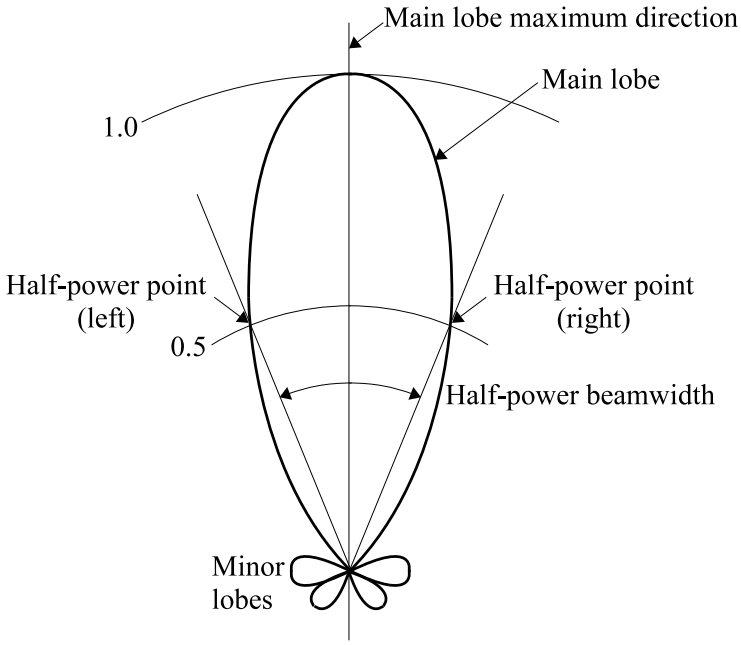


Figure 2.3: Typical antenna pattern polar plot.

2.2.1 Directivity

The directivity is a figure-of-merit describing how well the radiator directs energy in a certain direction. The directivity of an antenna is defined as the ratio of the

radiation intensity in a given direction from the antenna to the radiation intensity averaged over all directions. The average radiation intensity is equal to the total power radiated by the antenna divided by 4π. In other words, the directivity is the ratio of the power density of an anisotropic antenna relative to an isotropic antenna radiation in the same total power. If the direction is not specified, the direction of maximum radiation intensity is implied [21]. In mathematical form, it can be written as

$$D = \frac{U}{U_0} = \frac{4\pi U}{P_{rad}} \tag{2.1}$$

where

D = directivity (dimensionless)

U = power density (W/unit solid angle)

U_0 = radiation intensity of isotropic source (W/unit solid angle)

P_{rad} = total radiated power (W)

By substituting the radiation intensity in Equation 2.1, the directivity can be written as

$$D(\theta, \phi) = \frac{4\pi\ U(\theta, \phi)}{\int_0^{2\pi} \int_0^{\pi} U(\theta, \phi)\ sin(\theta)\ d\theta d\phi} \tag{2.2}$$

The maximum directivity, denoted by D_0 is a constant and is the maximum of Equation 2.2. Thus, the maximum directivity is obtained by calculating the maximum radiation intensity

$$D_0 = \frac{4\pi\ U_{max}}{\int_0^{2\pi} \int_0^{\pi} U(\theta, \phi)\ sin(\theta)\ d\theta d\phi} \tag{2.3}$$

In an isotropic element, the directivity is equal to 1 since they radiate equally in all directions and therefore are not directive. In addition to directivity, the radiation pattern of an antenna is also characterized by its beamwidth and side-lobe levels as discussed in the following subsections.

2.2.2 Half-Power Beamwidth

The *Half-Power Beamwidth* (HPBW) is defined as: "In a plane containing the direction of the maximum of a beam, the angle between the two directions in which the radiation intensity is one-half the maximum value of the beam." [20] The HPBW is measured from the 3-dB points of a radiation pattern. The HPBW is the angle between the 3-dB points. Since this is a power pattern, the 3-dB points are also the half power points. [16]. The beamwidth of the antenna is a very important figure-of-merit. The smaller the HPBW is, the easier it is to avoid interference from undesired signals.

2.2.3 Sidelobe Level

In most cases the main lobe is the intended lobe and thus the minor lobes are side lobes. A measure of how well the power is concentrated into the main lobe is the (relative) *Side-Lobe Level* (SLL) which is the ratio of the pattern value of a side lobe peak to the pattern value of the main lobe.

2.2.4 Array Factor

One of the most important functions in array theory is the *Array Factor* (AF). The array factor is a function of the positions of the antennas in the array and the weights used. By tailoring these parameters, the array performance may be optimized to achieve desirable properties. For instance, the array can be steered (change the direction of maximum radiation or reception) by changing the weights [22].

2.3 Antenna Array Geometry

The radiation pattern of a single antenna element is relatively wide and the values of directivity are normally low. Adaptive antenna arrays must be able to radiate power towards a desired angular sector to allow long distance transmissions and to avoid interference with undesired signals. One way to increase the gain is to enlarge the dimension of the antenna element but this could be a problem with

mobile devices due to their size. Another way to enlarge the dimension of the antenna system is to create a collection of two or more antennas in an electrical and geometrical configuration. This set of antenna elements is an *antenna array* and has a unique radiation pattern which is dictated by five factors:

- The geometrical configuration of the array
- The distance between individual elements
- The excitation phase of the individual elements
- The excitation amplitude of the individual elements
- The relative pattern of the individual elements

In the next subsections, different configurations of 2-dimensional antenna arrays will be studied.

2.3.1 Uniform Linear Array

The antenna elements placed along a line are the simplest of antenna array configurations. Let us assume that the antenna under investigation is an array of N isotropic radiating elements positioned along the x-axis equidistant from each other as shown in Figure 2.4. The total field of the array is equal to the field of a single element positioned at the origin multiplied by a factor which is widely referred to as the *array factor*. Thus, for the N-element array, the array factor is given by [16]

$$AF = 1 + \omega_1 e^{j(kd\ sin(\theta)+\beta_1)} + \omega_2 e^{j2(kd\ sin(\theta)+\beta_2)} + ... + \omega_N e^{j(N-1)(kd\ sin(\theta)+\beta_N)} \quad (2.4)$$

This can also be expressed as:

$$AF = \sum_{n=1}^{N} \omega_n e^{j(n-1)(kd\ sin(\theta)+\beta_n)} = \sum_{n=1}^{N} \omega_n e^{j(n-1)\psi} \quad (2.5)$$

where $\psi = kd\ sin(\theta) + \beta_n$ or if the array is aligned along the z-axis, $\psi = kd\ cos(\theta) + \beta_n$. k is the wavenumber and equals to $2\pi/\lambda$, λ being the wavelength.

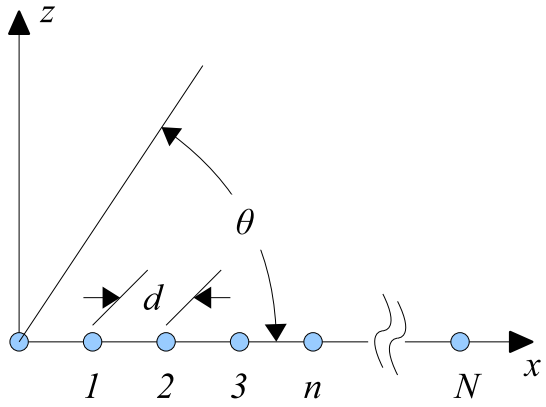


Figure 2.4: Linear array of N elements positioned along the x-axis.

d is the distance between elements. θ is the angle as measured from the y-axis in spherical coordinates. ω_n is the amplitude weight at element n and β_n is the phase shift weight at element n.

The total array factor is a summation of exponentials, thus it can be represented as a vector, which is called the *array vector* and is written as

$$\begin{aligned} \bar{a}(\theta) &= \begin{bmatrix} 1 \\ \omega_n e^{j(kd\ sin(\theta)+\beta_n)} \\ \vdots \\ \omega_n e^{j(N-1)(kd\ sin(\theta)+\beta_n)} \end{bmatrix} \\ &= \begin{bmatrix} 1 & \omega_n e^{j(kd\ sin(\theta)+\beta_n)} & \dots & \omega_n e^{j(N-1)(kd\ sin(\theta)+\beta_n)} \end{bmatrix}^T \end{aligned} \tag{2.6}$$

where $[]^T$ is the transpose of the vector within the brackets. The vector notation in Equation (2.6) will be used in Chapter 5.

2.3.2 Uniform Rectangular Array

A planar array consists of individual radiators positioned along a rectangular grid. Planar arrays are versatile and can provide more symmetrical patterns with lower side lobes compared to linear arrays. Applications include tracking radar, search radar, remote sensing, communications, and many others [23]. Figure 2.5 shows

a rectangular array in the x-y plane. There are M elements in the x-direction and N elements in the y-direction creating an $M \times N$ array of elements. The spaces between elements are d_x and d_y for the x-directed and y-directed elements respectively. Each element on the array has a weight ω_{mn}. A planar array is equivalent to M linear arrays of N elements or N linear arrays of M elements. The pattern of the entire $M \times N$ element array can be deduced by multiplying the array factors of the corresponding linear arrays as shown in Equation 2.7 [16].

$$\begin{aligned} AF &= AF_x \cdot AF_y \\ &= \sum_{m=1}^{M} a_m e^{j(m-1)(kd_x sin(\theta)cos(\phi)+\beta_x)} \\ &\cdot \sum_{n=1}^{N} b_n e^{j(n-1)(kd_y sin(\theta)sin(\phi)+\beta_y)} \end{aligned} \tag{2.7}$$

a_m and b_n being the amplitude weights. k is the wavenumber and equals to $2\pi/\lambda$, where λ is the wavelength. θ and ϕ are the angles as measured from the z-axis in spherical coordinates and β_x and β_y are the phase delays for beamsteering. The array factor can also be written as

$$AF = \sum_{m=1}^{M} \sum_{n=1}^{N} \omega_{mn} e^{j(m-1)\psi_x+(n-1)\psi_y} \tag{2.8}$$

where $\omega_{mn} = a_m \cdot b_n$ and is a set of complex array weights for each mn^{th} element. Finally, $\psi_x = (kd_x sin(\theta)cos(\phi) + \beta_x)$ and $\psi_y = (kd_y sin(\theta)sin(\phi) + \beta_y)$.

2.3.3 Uniform Circular Array

The circular array, in which the elements are placed in a circular ring, is an array configuration of very practical interest. Its applications span radio direction finding, air and space navigation, underground propagation, radar, sonar, and many other systems [24]. The array elements are placed on the $x-y$ plane forming

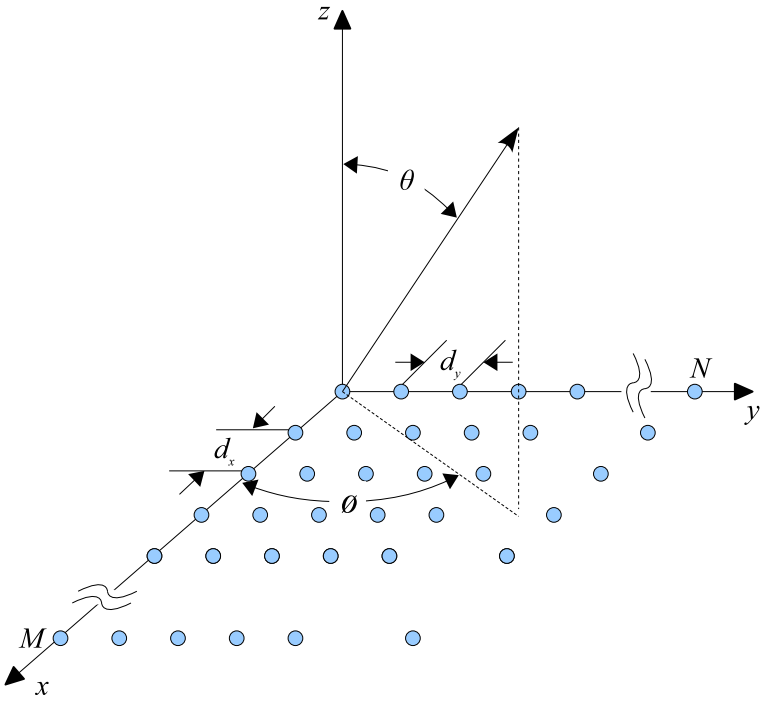


Figure 2.5: Rectangular array geometry.

a circle of radius a, and two angles, ϕ for azimuth and θ for elevation, represent the components of the desired direction. The circular array configuration is shown in Figure 2.6. The array factor of a circular array of N equally spaced elements is written as [24]

$$AF = \sum_{n=1}^{N} \omega_n e^{j[ka\ sin(\theta)\ cos(\phi-\phi_n)+\beta_n]} \tag{2.9}$$

where

N = number of isotropic antenna elements

$k = \frac{2\pi}{\lambda}$ = wavenumber

a = radius of the circular ring

ω_n = amplitude excitation of the n^{th} element

β_n = phase excitation of the n^{th} element

$\phi_n = 2\pi(\frac{n}{N})$ = angular position of the n^{th} element

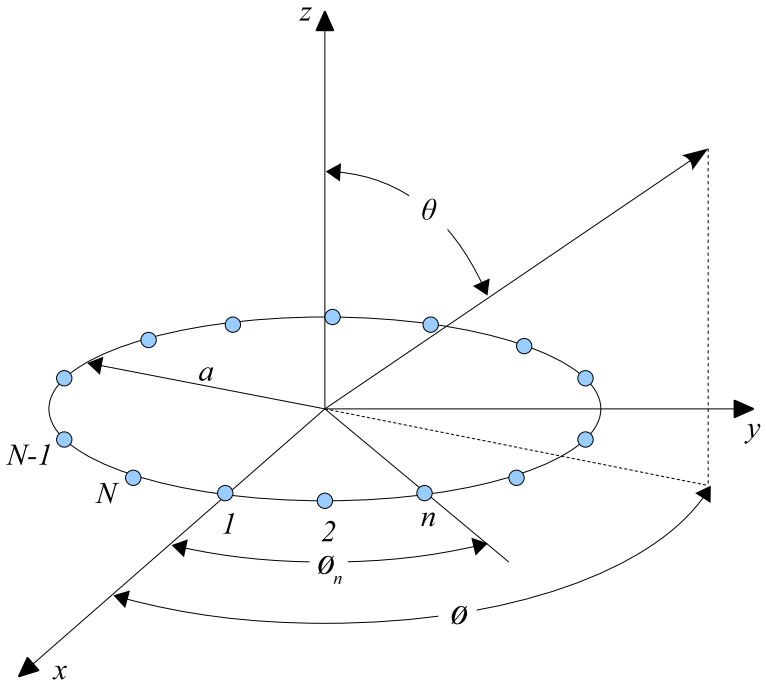


Figure 2.6: Circular array geometry.

2.4 Signal-to-Interference-plus-Noise Ratio

Maximizing the Signal-to-Interference-plus-Noise Ratio or SINR is a criterion which can be applied to enhancing the received signal while minimizing the interfering signals [25]. The SINR is defined as the ratio of the desired signal power divided by the undesired signal power and is given by Equation 2.10 [16]

$$SINR = \frac{P_{ss}}{P_{uu}} = a^2 \frac{\left|\bar{w}^H \cdot \bar{x}_s\right|^2}{\bar{w}^H \cdot \bar{R}_{uu} \cdot \bar{w}} \tag{2.10}$$

An optimization criterion proposed by Applebaum [26], consists in maximizing SINR. But a direct maximization of Equation 2.10 is not possible since neither a nor $\bar{R}_{uu}$ can be directly measured. However, as shown in [27], the equation can be recast as the maximization of the fitness function $f(\bar{w})$

$$f(\bar{w}) = \frac{\left|\bar{w}^H \cdot \bar{x}_s\right|^2}{\bar{w}^H \cdot \bar{R}_{xx} \cdot \bar{w}} \tag{2.11}$$

were ^{H} means transpose and $\bar{w}$ is the complex array of weights given by

$$\bar{w} = \{w_{mn}e^{j\beta_{mn}}; m = 1, ..., M; n = 1, ..., N\} \tag{2.12}$$

the amplitude and phase of the mn^{th} element are w_{mn} and β_{mn} respectively. $\bar{R}_{xx}$ is the array correlation matrix for the received signal and is equal to

$$\bar{R}_{xx} = \bar{R}_{uu} + \bar{R}_{ss} \tag{2.13}$$

where $\bar{R}_{ss}$ is the desired signal correlation matrix and $\bar{R}_{uu}$ is the undesired correlation matrix given by

$$\bar{R}_{uu} = \bar{R}_{ii} + \bar{R}_{nn} \tag{2.14}$$

$\bar{R}_{ii}$ being the correlation matrix for interferers and $\bar{R}_{nn}$ the correlation matrix for noise. Finally, $\bar{x}_s$ is a vector that represents the array factor as explained in the previous section and is given by

$$\bar{x}_s = e^{j(m-1)(kd_x sin\theta cos\phi)+(n-1)(kd_y sin\theta sin\phi)} \tag{2.15}$$

Equation 2.11 can be used as the fitness function for population-based optimization algorithms like the Particle Swarm Optimization and Genetic Algorithms.

2.5 Summary

This chapter introduced the concept of smart antennas as well as their importance in modern communication systems. Smart antennas are comprised of multiple antenna elements with the aim of optimizing the overall radiation pattern. These antenna systems are capable of responding to changes in the signal environment. Adaptive arrays can extract the desired signal and filter undesired signals and noise. This is achieved by radiating the power towards a certain direction. Until recent years, these systems had a high cost due to their complexity. Nowadays, the advances in technology and the commercialization of mobile communication devices have allowed smart antennas to be implemented in real life applications. In this chapter the fundamentals of antenna arrays were also discussed. Smart

antenna radiation patterns are controlled via algorithms that are based in certain criteria. These algorithms can maximize the signal-to-interference-plus-noise ratio, or minimize the variance or mean-square error. These is often achieved by using adaptive algorithms, which make these kind of systems referred as adaptive beamforming. Figures-of-merit such as directivity, half-power beamwidth, sidelobe level etc. have been explained. Furthermore, the basic antenna array configurations, linear, rectangular and circular, were described along with their mathematical descriptions. Finally, the signal-to-interference-plus-noise ratio, which will be used in the following chapters was described.

Chapter 3

Literature Review: Optimization Methods

This chapter provides the reader with a general overview of the optimization algorithms which will be used in this thesis. The Genetic Algorithm as well as the Particle Swarm Optimization are used in Chapters 5 and 6 to optimize the radiation pattern of linear and rectangular antenna arrays. The remaining algorithms: Differential Evolution, Simulated Annealing, Hill Climb, Pattern Search and Local Unimodal Sampling are used in Chapter 7 for meta-optimization.

3.1 Introduction

Modern communication technologies have grown at its fastest pace in the last decade. The increasingly number of mobile devices used in the networks has lead to problems that have to be solved. As mentioned in the previous chapter, the use of adaptive antenna arrays in mobile communications can help to tackle problems like co-channel an multi-access interference. For an antenna array system to be smart, adaptive algorithms have to be applied in order to control and configure the system behaviour to constant changes in the environment. The methods used for operating adaptive antenna arrays can be broadly classified into two groups: deterministic and stochastic. The deterministic methods include analytical methods like fast fourier transform and least square methods [28]. These

methods are often computationally time consuming when the number of antenna elements is high. On the other hand, stochastic methods have some advantages over deterministic methods as explained in [29]. Evolutionary methods like Genetic Algorithm (GA) [30, 31], Simulated Annealing (SA) [32, 33, 34], Differential Evolution (DE) [35, 36] and Particle Swarm Optimization (PSO) [10] have the ability to deal with large number of dimensions and are often easily implemented on computers. The effectiveness of these algorithms for the design and operation of antenna arrays has lead the present work to study and use them to obtain and compare results in order to assess their suitability to solve the complex problems of array adaptation. The following sections present an overview of the optimization algorithms used throughout this thesis.

3.2 Genetic Algorithm

A Genetic Algorithm (GA) is a multi-agent optimization method inspired by the evolution of biological individuals that adapt to their environment through generations and mutation; a theory proposed by Darwin [37]. The use of GAs for numerical optimization is attributed to Holland [30] back in the 70s who proposed the use of this algorithmic approach to solve practical problems rather than for simulating biological systems. Another text on GAs is attributed to Goldberg [31]. This idea was rapidly accepted and spread. However, Holland's aim was to create a general framework for a kind of adaptive systems rather than to solve application specific problems. GAs begin with a set of x randomly generated states which are called *population*. A string over a finite alphabet represents each state which are called *chromosomes*, commonly, a string of 0s and 1s. The selection of the next generation of individuals depends on the evaluation of a fitness function. This function returns higher values for better individuals. The next step is to randomly select two pairs of individuals for reproduction. For each pair to be mated, a *crossover* point is chosen at random from the positions in the string. Then, the offspring themselves are created by crossing over the parent strings at the crossover point. For example, the first child of the first pair gets the first three digits from the first parent and the remaining digits from the second

parent, whereas the second child gets the first three digits from the second parent and the rest from the first parent. In the final step, each location is subject to random *mutation* with a small independent probability. For example, one digit is mutated in the first, third and fourth offspring. Genetic algorithms combine an uphill tendency with a random exploration and exchange of information amongst parallel search threads. Although it can be shown mathematically that, if the positions of the genetic code is permuted initially in a random order, crossover conveys no advantage. Genetic algorithms are good to solve problems that deal with the optimization of nonlienar multimodal functions that have many variables. Experimental results have shown that GAs are able to find good solutions to antenna systems. [38]. The pseudocode is shown in Algorithm 1.

3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a biologically-inspired optimization technique. It was proposed by Eberhart and Kennedy in 1995 [10]. PSO is inspired by the social behaviour of swarms of bees. In these biological systems, the collective behaviour of simple individuals in their environment leads to the solution of a given problem, for example, finding food. The goal is to find the location with the highest density of flowers by randomly flying over the field. Each bee can remember the location where it found the most flowers, and by dancing in the air, they communicate this information to other bees. Occasionally, one bee may fly over a place with more flowers than had been discovered by any bee in the swarm. Over time, more bees end up flying closer and closer to the best patch in the field. Soon, all the bees swarm around this point.

As an optimization technique, the system is initialized with a population of random solutions (also called *particles*) and searches for optima by updating generations. Each particle remembers its best solution called personal best or $\vec{p}$ and the global best or $\vec{g}$ which is the best solution achieved so far by any of the individuals. At each iteration, the particles update their velocity towards the $\vec{p}$ and $\vec{g}$ locations according to the following two equations:

Algorithm 1 GA algorithm.

```
1: Initialize the population randomly in the search-space.
2: while The termination criterion is not met do
3:    for Each individual in the population do
4:       Select two parents
5:       x ← RandomSelection(population, FitFunction)
6:       y ← RandomSelection(population, FitFunction)
7:       Reproduce both parents according to a defined crossover probability
8:       child ← Reproduce(x, y)
9:       if Small random probability then
10:         Mutate child
11:         child ← Mutation(child)
12:      end if
13:   end for
14:   Evaluate population
15:   population ← new − population
16: end while
```

$$\vec{v}_{n+1} = \omega \cdot \vec{v}_n + c_1 r_1(\vec{p}_n - \vec{x}_n) + c_2 r_2(\vec{g}_n - \vec{x}_n) \tag{3.1}$$

$$\vec{x}_{n+1} = \vec{x}_n + \vec{v}_{n+1} \tag{3.2}$$

where v_n and $\vec{x}_n$ are the particle velocity and position at the nth generation respectively. ω is the inertia weight and is used to control the trade-off between the global and the local exploration ability of the group of particles or *swarm*, usually in the range of [0,1]. c_1 and c_2 are scaling constants that determine the relative pull of $\vec{p}$ and $\vec{g}$, usually taken as $c_1 = c_2 = 2.0$. r_1 and r_2 are random numbers uniformly distributed in (0,1). Once the velocity has been calculated, the particle moves to its next location. The new coordinate is determined according to Equation 3.2. The swarm will continue moving until a criterion is met, usually a sufficiently good fitness value or a maximum number of iterations. The pseudocode for the PSO is shown in Algorithm 2. Unlike GAs, the PSO is based upon the cooperation amongst the individuals rather than their competition. In addition, it is easier to calibrate and to control the parameters of the PSO over the GA [39]. GAs require a specific strategy and careful choice of operators according to the application, whereas PSO eliminates the process of selecting the best operators by sequentially updating its equations.

3.4 Differential Evolution

In 1995, Price and Storn proposed a multi-agent heuristical optimization method called Differential Evolution (DE) [35, 36]. Differential Evolutions grew out of attempts to solve the Chebychev Polynomial fitting Problem. A breakthrough came when Price came up with the idea of using vector differences for perturbing the vector population. DE basically works by creating a new possible agent-position by combining the position of randomly chosen agents from its population, and updating the agent's current position in case there is improvement to the fitness. In other words, instead of classical crossover or mutation, it creates new offspring from parent chromosomes by using a differential operator. Since the publication of Price and Storn, the Differential Evolution algorithm has been through sub-

Algorithm 2 PSO algorithm.

1: Initialize the particles with random velocities and random positions in the search-space.

2: **while** The termination criterion is not met **do**

3: **for** Each particle in the swarm **do**

4: Pick two random numbers: $\vec{r_1}$, $\vec{r_2} \sim U(0,1)$.

5: Update the particle's velocity $\vec{v}$ as follows:

6: $\vec{v}_{n+1} = \omega \cdot \vec{v}_n + c_1 r_1(\vec{p}_n - \vec{x}_n) + c_2 r_2(\vec{g}_n - \vec{x}_n)$

Where $\vec{g}$ is the swarm's best known position, $\vec{p}$ is the particle's own best known position, and ω, c_1 and c_2 are user-defined behavioural parameters.

7: Move the particle to its new position by adding its velocity:

8: $\vec{x} \leftarrow \vec{x} + \vec{v}$

9: **if** $f(\vec{x}) < f(\vec{p})$ **then**

10: Update the particle's best known position:

11: $\vec{p} \leftarrow \vec{x}$

12: **end if**

13: **if** $f(\vec{x}) < f(\vec{g})$ **then**

14: Update the swarm's best known position:

15: $\vec{g} \leftarrow \vec{x}$

16: **end if**

17: **end for**

18: **end while**

stantial improvement which make it a versatile and robust tool. It is worth noting that DE managed to finish 3rd at the First International Contest of Evolutionary Computation (ICEO) held in Nagoya, May 1996 [40], where the first two places were given to non-GA type algorithms which are not universally applicable but which solved the test-problems faster than DE. Like Genetic Algorithms, DE also employs operators that are dubbed crossover and mutation but with different meanings. In this thesis, the classic DE (DE/rand/1/bin) will be used as it is believed to be the best performing and hence most popular of the DE variants [41] There are several other variations, for example the JDE Variant proposed by Brest et al. [42]. DE starts with a population of NP D-dimensional search variable vectors. Subsequent generations are presented by discrete time steps $(t = 0, 1, 3, ..., t, t + 1,)$ etc. As the vectors can change over different generations, the following notation for representing the ith vector is adopted

$$\vec{X}_i(t) = [x_{i,1}(t),\ x_{i,2}(t),\ x_{i,3}(t),\ \dots\ , x_{i,D}(t)] \tag{3.3}$$

where $\vec{X}_i(t)$ are vectors called "genomes" or "chromosomes". For each variable, there may be a certain range within which the value of the parameter should lie for better search results. At the beginning, or at $t = 0$, the problem parameters are initialized within a defined range. Therefore, if the jth parameter of the given problem has lower and upper bounds x_j^L and x_j^U respectively, then the jth component of the ith population member is initialized as follows

$$x_{i,j}(0) = x_j^L + U(0,1) \cdot \left(x_j^U - x_j^L\right) \tag{3.4}$$

where $U(0, 1)$ is a uniformly distributed random number lying between 0 and 1. In each generation, to change each population member $\vec{X}_i(t)$, a donor vector $\vec{V}_i(t)$ is created. To create a $\vec{V}_i(t)$ for each ith member, three other parameter vectors $\vec{a}$, $\vec{b}$ and $\vec{c}$ are chosen randomly from the current population. Next, a scalar number F called the differential weight, scales the difference of any two of the three vectors and the scaled difference is added to the third one so the $\vec{V}_i(t)$ vector is obtained. This process can be expressed for the jth component of each

vector as [35]

$$v_{i,j}(t+1) = x_{\vec{a},i}(t) + F \cdot \left(x_{\vec{b},i}(t) - x_{\vec{c},i}(t) \right) \tag{3.5}$$

Next, to improve the potential diversity of the population, a crossover scheme is applied. DE can use two kinds of crossover schemes namely "Exponential" and "Binomial". The donor vector exchanges its components with the target vector $\vec{X}_i(t)$. In "Exponential" crossover, a random integer n is chosen among the interval $[0, D-1]$. This integer is the starting point in the target vector, from where the crossover or exchange of components with the donor vector will take place. Another integer L is chosen from the interval $[1, D]$ to represent the number of components contributed by the donor vector to the target. Once n and L are chosen the trial vector [35]

$$\vec{U}_i(t) = [u_{i,1}(t),\ u_{i,2}(t),\ x_{i,3}(t),\ \ldots\ , u_{i,D}(t)] \tag{3.6}$$

is formed with

$$\begin{aligned} u_{i,j}(t) &= v_{i,j}(t) \quad \text{for} \quad j = < n >_D,\ < n+1 >_D,\ \ldots\ , < n-L+1 >_D \\ &= x_{i,j}(t) \end{aligned} \tag{3.7}$$

where the angular brackets $<>_D$ denote a modulo function with modulus D. The integer L is drawn from $[1, D]$ according to the following pseudocode

$L = 0$;
while $(U(0,1) < CR)$ AND $(L < D))$ **do**
  $L = L + 1$;
end while

Hence in effect probability $(L > m) = (CR)^{m-1}$ for any $m > 0$. CR is the crossover probability and is one of the main control parameters of DE just like F. For each donor vector V, a new set of n and L must be chosen randomly as shown above. However, in the "Binomial" crossover scheme, the crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1

is within the CR value. The scheme can be outlined as

$$u_{i,j}(t) = \begin{cases} v_{i,j}(t) & \text{, if } U(0,1) < CR \\ x_{i,j}(t) & \text{, else} \end{cases} \tag{3.8}$$

In this way, for each trial vector $\vec{X}_i(t)$ an offspring vector $\vec{U}_i(t)$ is generated. To keep the population size constant over subsequent generations, the next stem of the algorithm is a "selection" to determine which one of the target vector and the trial vector will survive in the next generation. DE actually involves the Darwinian principle of "Survival of the fittest" in its selection process which may be outlined as

$$\vec{X}_i(t+1) = \begin{cases} \vec{U}_i(t) & \text{, if } f(\vec{U}_i(t)) \leq f(\vec{X}_i(t)) \\ \vec{X}_i(t) & \text{, if } f(\vec{X}_i(t)) < f(\vec{U}_i(t)) \end{cases} \tag{3.9}$$

where $f()$ is the function to be minimized. So if the new trial vector obtains a better value of the fitness function, it replaces its target in the next generation, otherwise the target vector is retained in the population. Thus, the population either gets better or it remains constant but never deteriorates. The DE algorithm is shown as a pseudocode in Algorithm 3.

3.5 Simulated Annealing

A hill climb algorithm that never makes downhill moves towards states with lower value will never be complete because it can get stuck on a local maximum. On the other hand, a purely random walk that moves to a successor chosen uniformly at random from the set of successors is complete but can be extremely inefficient. Thus, it seems reasonable to try to combine both hill climb with a random walk in some way that yields both efficiency and completeness [43]. Simulated annealing (SA) is a generic and probabilistic meta-heuristic algorithm, introduced in the 80s by Kirkpatrick et al. [32, 33, 34]. While genetic algorithms are biologically inspired, simulated annealing is "metallurgy inspired". Thermal annealing is a technique involved in metallurgy to reduce the defects of a material

Algorithm 3 DE algorithm.

1: Initialize the agents with random positions in the search-space

2: **while** The termination criterion is not met **do**

3: **for** Each agent $v_{i,j}(t)$ in the population **do**

4: Pick three agents $\vec{a}$, $\vec{b}$ and $\vec{c}$ at random, they must be distinct from each other as well as from agent $v_{i,j}(t)$.

5: Compute the agent's potentially new position $u_{i,j}(t)$, by iterating over each $i \in \{1, ..., n\}$ as follows:

- Pick $U(0, 1)$ for use in a stochastic choice next.
- Compute the ith element of the potentially new position $u_{i,j}(t)$, using Equation 3.8 from above:
- $$u_{i,j}(t) = \begin{cases} v_{i,j}(t) & \text{, if } U(0,1) < CR \\ x_{i,j}(t) & \text{, else} \end{cases}$$
- Where the user-defined behavioural parameters are the differential weight F and the crossover probability CR.

6: **if** $f(\vec{U}_i(t)) < f(\vec{X}_i(t))$ **then**

7: Update the agent's position:

8: $\vec{X}_i(t+1) \leftarrow \vec{U}_i(t+1)$

9: **end if**

10: **end for**

11: **end while**

by heating and controlled cooling. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. By analogy with this physical process, each step of the SA algorithm replaces the current solution with a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter T (called the temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly downhill as T goes to zero. The allowance for uphill moves saves the method from becoming stuck at local minima, which are the bane of greedier methods. The pseudocode of the SA algorithm is shown in Algorithm 4. The algorithm starts by generating an initial solution (usually a random solution) and by initializing the so-called temperature parameter T. Then the following is repeated until the termination condition is satisfied: a solution y from the neighbourhood of the current solution is randomly sampled and it is accepted as a new current solution if

$$Fitness(Solution) < Fitness(\vec{y}) \tag{3.10}$$

If this condition is not met, the probability decreases exponentially according to the temperature:

$$e^{\left(\frac{(-f(Solution)-f(\vec{y})}{T}\right)} \tag{3.11}$$

3.6 Hill Climb

The Hill Climb (HC) is an optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a function and attempts to find a better solution by incrementally changing a single variable of the solution [43, 44, 45]. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found. For example, hill climb can be applied to the

Algorithm 4 SA algorithm.

1: Initialize the agents with random positions in the search-space.
2: Initialize temperature $T \leftarrow T_0$
3: **while** The termination criterion is not met ($T \leftarrow 0$) **do**
4: $\vec{y} \leftarrow rand(\vec{x})$
5: Choose randomly from $\vec{x}$
6: **if** $f(\vec{y}) < f(\vec{x})$ **then**
7: $Solution \leftarrow \vec{y}$
8: Update the *Solution* value with the better one
9: **else**
10: $Solution \leftarrow \vec{y}$ with *probability*
11: Update the *Solution* value with a new probability: $e^{\left(\frac{(-f(Solution)-f(\vec{y})}{T}\right)}$
12: Update the temperature T
13: **end if**
14: **end while**

travelling salesman problem. It is easy to find an initial solution that visits all the cities but it will be very poor compared to the optimal solution. The algorithm starts with such a solution and makes small improvements to it, such as switching the order in which two cities are visited. Eventually, a much shorter route is likely to be obtained. Hill Climb is good for finding a local optimum (a good solution that lies relatively near the initial solution) but it is not guaranteed to find the best possible solution (the global optimum) out of all possible solutions (the search space). The relative simplicity of the algorithm makes it a popular first choice amongst optimizing algorithms. It is used widely in artificial intelligence, for reaching a goal state from a starting node. The choice of next node and starting node can be varied to give a list of related algorithms. Although more advanced algorithms such as simulated annealing or tabu search may give better results, in some situations Hill Climb works just as well. Algorithm 5 shows the HC algorithm.

3.7 Pattern Search

Pattern Search (PS), described by Hooke and Jeeves [46], is a family of numerical optimization methods that samples the search-space locally from the current position and decreases its sampling-range upon failure to improve its fitness [47]. PS does not require the gradient of the problem to be optimized and can hence be used on functions that are not continuous. An early and simple PS variant is attributed to Fermi and Metropolis when they worked at the Los Alamos National Laboratory as described by Davidon [48]. One of the theoretical parameters is varied at a time by steps of the same magnitude, and when no such increase or decrease in any one parameter further improved the fit to the experimental data, the step size is halved and the process repeated until the steps are smaller than on the desired threshold. The idea of PS is similar to that of Golden Section Search (GSS) by Kiefer [49], which works for one-dimensional search-spaces by maintaining three separate points, and at each iteration replacing one of these with an intermediate point that is chosen so as to close in on the optimum of an unimodal problem. The variant presented in this thesis is the one used by

Algorithm 5 HC algorithm.

1: Initialize the neighbours with random positions in the search-space.
2: Initialize $MaxFitness$ which is the maximum fitness reached at a point.
3: $Found \leftarrow true$
4: **while** The termination criterion is not met ($Found \leftarrow true$) **do**
5: $Found \leftarrow false$
6: **for** Each neighbour $\vec{x}$ **do**
7: **if** $f(\vec{x}) > MaxFitness$ **then**
8: $MaxFitness \leftarrow f(\vec{x})$
9: Update the $MaxFitness$ value with the new one
10: $Solution \leftarrow \vec{x}$
11: Update the overall solution $Solution$
12: $Found \leftarrow true$
13: **end if**
14: **end for**
15: **end while**

Pedersen in [50]. The pseudocode of PS is shown in Algorithm 6.

3.8 Local Unimodal Sampling

Local Unimodal Sampling (LUS) is an optimization method which can be thought of as an extension of the previously discussed PS method. It was introduced in 2008 by Pedersen [51]. It samples all dimensions simultaneously, while decreasing its sampling-range in much the same manner as PS. The reason for decreasing the sampling-range during optimization is that a fixed sampling-range has no possibility of converging to a local optimum [50]. LUS decreases the search-range exponentially when samples fail to improve on the fitness of the current position. Some optimization techniques use exponential decrease of search-range, for example the Luus-Jaakola method [52] and also the method presented by Fermi and Metropolis [48]. For the sampling done by the LUS method, the new potential position denoted by $\vec{y}$ is chosen from the neighbourhood of the current position $\vec{x}$:

$$\vec{y} = \vec{x} + \vec{a} \tag{3.12}$$

where the vector $\vec{a}$ is randomly and uniformly generated:

$$\vec{a} \sim U\left(-\vec{d}, \vec{d}\right) \tag{3.13}$$

where $\vec{d}$ is the current sampling-range, initially chosen as the full range of the search-space and decreased during optimization. When a sample fails to improve the fitness, the sampling-range is decreased for all dimensions simultaneously. The amount by which the sampling-range will be decreased is calculated in the same way as in the PS method. The sampling-range is halved for every dimension after n failures to improve fitness. The sampling-range $\vec{d}$ should therefore be multiplied with q for each failure to improve the fitness:

$$\vec{d} = q \cdot \vec{d} \tag{3.14}$$

with q being defined as:

$$q \leftarrow \sqrt[n]{1/2} \tag{3.15}$$

Algorithm 6 PS algorithm.

1: Initialize $\vec{x}$ to a random position in the search-space:

$\vec{x} \sim U\left(\vec{b}_{lo}, \vec{b}_{up}\right)$

where $\vec{b}_{lo}$ is the lower boundary of the search-space and $\vec{b}_{up}$ is the upper boundary.

2: Set the initial sampling range $\vec{d}$ to cover the entire search-space:

$\vec{d} \leftarrow \vec{b}_{up} - \vec{b}_{lo}$

3: **while** The termination criterion is not met **do**

4: Pick an index $R \in \{1, ..., n\}$ uniformly and randomly

5: Let $\vec{y}$ be the potentially new position in the search-space, wich is exactly the same as the current position $\vec{x}$, except for the Rth element y_R, which is found from the neighbourhood of x_R simply by adding d_R:

$$y_i = \begin{cases} x_i + d_i & , i = R \\ x_i & , else \end{cases}$$

6: **if** $f(\vec{y}) < f(\vec{x})$ **then**

7: Keep the new position: $\vec{x} \leftarrow \vec{y}$

8: Otherwise update the sampling-range and direction for the Rth dimension: $d_R \leftarrow -\frac{d_R}{2}$

9: **end if**

10: **end while**

where n is the dimensionality of the problem to be optimized. The LUS algorithm is shown in Algorithm 7

3.9 Summary

In this chapter, the optimization algorithms used throughout this thesis were presented. Each algorithm was discussed and a general overview as well as pseudocode was explained. Evolutionary methods have been selected to solve antenna problems as they are capable of dealing with large number of dimensions. In the case of antenna arrays, each antenna element added to the system represents an additional dimension that has to be solved. The Genetic Algorithm combines an uphill tendency with a random exploration and exchange of information amongst parallel search threads. Moreover, Genetic algorithms are good to solve problems that deal with the optimization of nonlienar multimodal functions that have many variables. Experimental results have shown that GAs are able to find good solutions to antenna systems [38]. On the other hand, unlike GAs, the PSO is based upon the cooperation amongst the individuals rather than their competition. In addition, it is easier to calibrate and to control the parameters of the PSO over the GA [39]. GAs require a specific strategy and careful choice of operators according to the application, whereas PSO eliminates the process of selecting the best operators by sequentially updating its equations. For these reasons, the GA and PSO algorithms have been chosen in this work to tackle antenna array problems as will be discussed in Chapters 5 and 6.

Algorithm 7 LUS algorithm.

1: Initialize $\vec{x}$ to a random position in the search-space:

$\vec{x} \sim U\left(\vec{b}_{lo}, \vec{b}_{up}\right)$

where $\vec{b}_{lo}$ is the lower boundary of the search-space and $\vec{b}_{up}$ is the upper boundary.

2: Set the initial sampling range $\vec{d}$ to cover the entire search-space:

$\vec{d} \leftarrow \vec{b}_{up} - \vec{b}_{lo}$

3: **while** The termination criterion is not met **do**

4: Pick a random vector $\vec{a} \sim U\left(-\vec{d}, \vec{d}\right)$

5: Add this to the current position $\vec{x}$, to create the new potential position $\vec{y}$: $\vec{y} = \vec{x} + \vec{a}$

6: **if** $f(\vec{y}) < f(\vec{x})$ **then**

7: Update the new position: $\vec{x} \leftarrow \vec{y}$

8: Otherwise decrease the sampling-range by the factor q from Equation 3.15: $\vec{d} \leftarrow q \cdot \vec{d}$

9: **end if**

10: **end while**

Chapter 4

Antenna Array Geometry

This chapter analyses the effect of a central antenna element on the radiation pattern in a uniform circular antenna array. A modification of the array geometry is considered in which one of the antenna elements is placed in the centre of the array. The corresponding array factor is adjusted to describe the geometric configuration that includes the central antenna element. This distribution alters the radiation pattern in such a way that the array directivity and half-power beamwidth are affected. An increase on the directivity and a decrease of the half-power beamwidth are obtained by adjusting the phase of the central element. A reduction of the side-lobe levels is also achieved. Array configurations with different number of antenna elements were also tested, and the results on directivity and half-power beamwidth are presented. Using Microstripes, a software tool that enables the simulation of antennas, a 6-element circular antenna array was designed and the directivity for a range of frequencies was obtained. Moreover, additional results were obtained for a range of transmission frequencies.

4.1 Introduction

Since the beginning of the twentieth century, antenna designers have investigated different antenna architectures to meet the requirements of communication systems. Nowadays, these efforts can benefit from the use of simulation software tools which allows the exploration of a large variety of configurations before fab-

rication, thus, reducing design times, costs etc. A large variety of antennas have been developed to date [23, 26, 53, 54]; they range from simple structures such as monopoles and dipoles to complex structures such as phased arrays. Usually, the radiation pattern of a single element antenna is very wide and it provides low values of directivity. As noted in Chapter 2, this problem can be solved by increasing the size of the antenna to obtain higher values of gain. However in the case of mobile devices, a higher size implies expenditure of energy which is normally very restrained. Another way to enlarge an antenna is to consider a set of individual elements in a geometrical configuration creating an antenna array [22]. It is convenient that the elements of the array are identical to enable a simpler analysis and design. The overall radiation pattern of the array is obtained with the phasor summation of each radiated field of every individual element. The interaction among all radiation patterns depends on the geometry of the array (number of elements, distance between elements, etc.) where the pattern of the elements should interfere constructively in the direction of the Signals-Of-Interest (SOI) and destructively in any other direction or Signals-Not-Of-Interest (SNOI). To determine the shape of the radiation pattern, five characteristics of the array can be adjusted [20]:

- The geometry configuration of the array (linear, rectangular, circular, etc)
- The excitation phase of the individual elements
- The excitation amplitude of the individual elements
- The relative displacement between the elements
- The relative pattern of the individual elements

The excitation phase and amplitude has received extensive attention [55, 56, 57]. However, the array geometry has received relatively little attention even though it also strongly influences the radiation pattern. The reason for this is primarily due to the complex way in which the geometry affects the radiation pattern [58]. Numerous studies for different geometries have been conducted in the past [59, 60]. However, these studies include mostly uniform linear arrays (ULAs)

and uniform rectangular arrays (URAs). In [4, 5], the performance of uniform circular arrays (UCAs) was examined. It was found that this arrangement of elements have no edge constraints so the beam pattern can be electronically rotated. Circular arrays have also the capability of compensating the effect of mutual coupling by breaking down the array excitation into a series of symmetrical spatial components. In other words, the symmetry of UCAs provides a major advantage when scanning a beam pattern azimuthally through 360° with little change in either beamwidth or sidelobe level.

Other work in antenna array geometry has been done in [61], where the capacity of different array configurations was studied. The capacities of these configurations were measured in terms of SNR. Also in this field, several antenna array geometries on MIMO channel eigenvalues were investigated in [62]. Four different antenna array geometries were considered, namely, uniform linear array, uniform circular array, uniform rectangular array and uniform cubic array. All the considered geometries had the same number of elements and fixed inter-element spacing. The uniform linear array geometry showed superiority to the other considered geometries.

Not only can the number of elements in a circular array be varied but the actual position in the ring can be carefully selected to obtain a desired radiation pattern. In [63] an optimum antenna array geometry was obtained in terms of suppressing interference. An optimization algorithm, namely Simulated Annealing was used to find the optimum array positioning and several configurations were presented. In [64], a genetic algorithm was used to optimize the element placement in a concentric ring array to obtain the lowest maximum sidelobe level at boresight. This optimization found the spacing that balances the height for all the sidelobes. It has also been observed that a planar arrangement with an element at the centre increases array steering capability as well as reducing the side lobe levels [65].

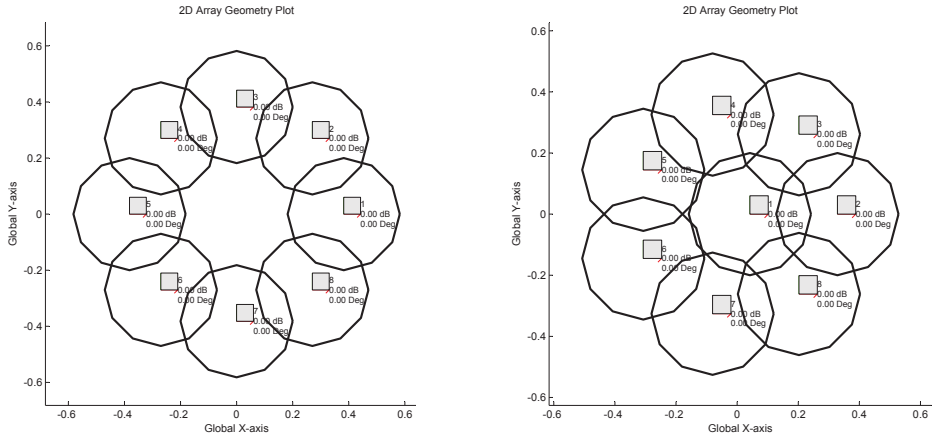
4.2 Circular Array with Central Element

Previous studies have pointed out the advantages of circular antenna arrays as well as the observations made in [65] of a central element. Uniform Circular Arrays, or (UCA) are a popular type of antenna arrays which have several advantages such as scan capability (they can perform a 360° scan) while the beam pattern is kept invariant [66, 67]. For these reasons, the present work contemplates the use of Uniform Circular Arrays and explores a variation of the geometry structure. The circular configuration is rearranged in such a way that one of the elements is placed in the centre of the array. As shown by the results in this chapter, this modification permits better values of directivity and half-power beamwidth compared to the original UCA.

This arrangement of emitters has been proposed in the past [68]: SPEAR is an antenna design consisting of one central element connected to the source and several surrounded parasitic elements in a circle. By adjusting the value of the reactance, the parasitic elements form the antenna array radiation pattern into different shapes. In the present work, different shapes of the radiation pattern are obtained by using different excitation phase angles for the central element. Additionally, this work includes comparisons between the standard and the modified UCA designs. Results for a range of different frequencies against directivity between both architectures are also presented. In addition, a circular array with a central element is presented in [69], where spiral elements are arranged in an hexagonal shape. This work obtains total gain results but the number of elements is fixed. In the present work, several UCA arrays with different number of elements (from 4 to 20) are explored to compare and obtain the best array configuration. Figure 4.1 shows both the standard and modified arrays studied in this thesis.

4.3 Deduction of Radiation Pattern Formula

As presented in Chapter 2, the array factor for a circular array of N equally spaced elements is [16]



(a) Standard UCA. (b) Modified UCA.

Figure 4.1: Standard and Modified uniform circular arrays.

$$AF = \sum_{n=1}^{N} \omega_n e^{j[ka\ sin(\theta)\ cos(\phi - \phi_n) + \beta_n]} \tag{4.1}$$

And to steer the main lobe in the (θ_0, ϕ_0) direction, the phase excitation of the nth element can be chosen to be

$$\beta_n = -ka \sin\theta_0 \cos(\phi_0 - \phi_n) \tag{4.2}$$

Given that the modified array shown in Figure 4.1(b) has one antenna element at the centre and the radius for this element is 0, the displacement phase factor on the array factor becomes $e^{j\beta_x}$ where β_x is the phase excitation of the element at the centre. The total field of the array is determined by the addition of the fields radiated by the individual elements. Thus, the resulting array factor for the modified array is the sum of the array factor of the standard circular array plus the antenna element at the centre

$$AF(\theta, \phi) = e^{j\beta_x} + \sum_{n=1}^{N} I_n e^{j[ka\ \sin\theta\ \cos(\phi - \phi_n) + \beta_n]} \tag{4.3}$$

This array factor represents the modified circular antenna array shown in Figure 4.1(b) and will be used in the following section to obtain data and compare

its performance against the standard circular antenna array 4.1(a).

4.4 Results

In order to analyse the effect caused by a central element on the radiation pattern of a uniform circular antenna array, several experiments are performed. These experiments consist in a comparison between the standard and the modified UCA arrays. The first experiment measures both the directivity and the half-power beamwidth of circular arrays with different number of antenna elements. The second experiment explores the possibility of changing the phase of the central element. In practice, this shift in the phase can be obtained electronically and can affect the overall radiation pattern of the array. A third experiment consists in replicating the previous tests but this time using a 3D electromagnetic simulation tool called Microstripes [70]. The Microstripes software allows to design antenna arrays and performs simulations in terms of frequency response. These simulations show 3D representations of the beam pattern, which are also studied in this experiments.

4.4.1 Experiment 1: Directivity and HPBW

A comparison between the standard and modified UCAs is presented. Matlab simulations were performed to calculate the directivity based on the fields above the $x-y$ plane and the HPBW at the maxima. The directivity and HPBW were obtained for arrays with different number of antenna elements.

Figure 4.2 shows that the directivity of the modified UCA is higher than the one for the standard UCA in a range of 4 to 20-element antennas. The maximum value of directivity, 8.25dB, is obtained with a 6-element array. Then it gradually decreases if more antenna elements are used. This is due to the decrease of power provided by the single central element compared with the total power supplied by the rest of the elements. It should be noted that, up to 20 elements, the directivity of the modified UCA is still higher than the one of the standard UCA.

In the case of the HPBW (Figure 4.3), the modified UCA presents a smaller angle compared with the standard UCA for any number of antennas from 4 to

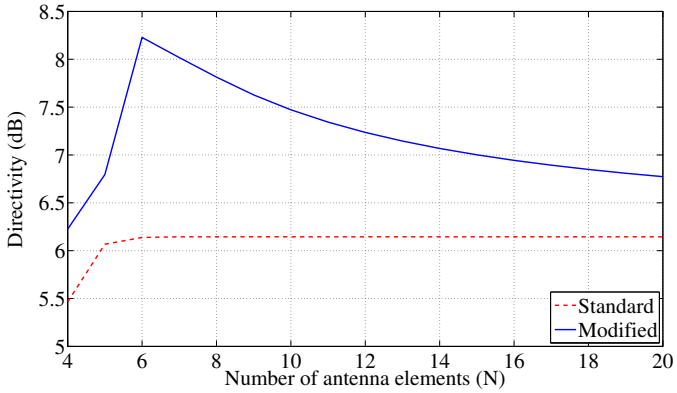


Figure 4.2: Directivity and number of elements.

20. This is a desired result given that a narrower HPBW allows the antenna to avoid Signals-Not-Of-Interest more effectively [54].

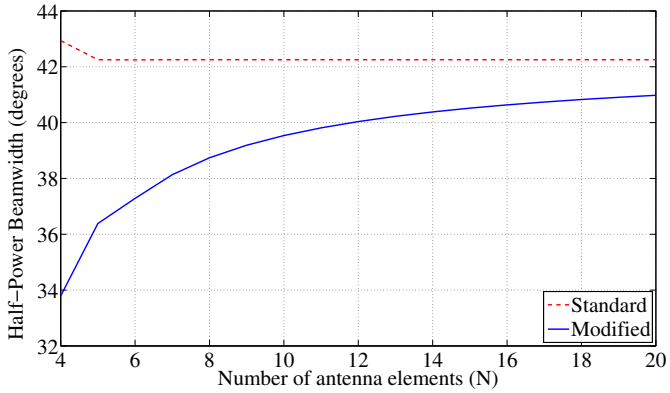


Figure 4.3: HPBW and number of elements.

4.4.2 Experiment 2: Phase shift of the central element

The second experiment consists in modifying the phase shift of the central element and obtain values of directivity and HPBW. The central element phase ϕ is shifted from 0°to 360°. Figures 4.4 and 4.5 show the changes on directivity and HPBW respectively. On both figures, the standard UCA (dashed line) shows a constant directivity and HPBW since it has no central element. It can be observed that with a phase shift of 180°, the directivity reaches its highest, while

the HPBW exhibits its smallest angle. Although, these values are lower than in the standard UCA for angles from 0°to 110°and 250°to 360°for directivity and from 0°to 100°and 260°to 360°for HPBW.

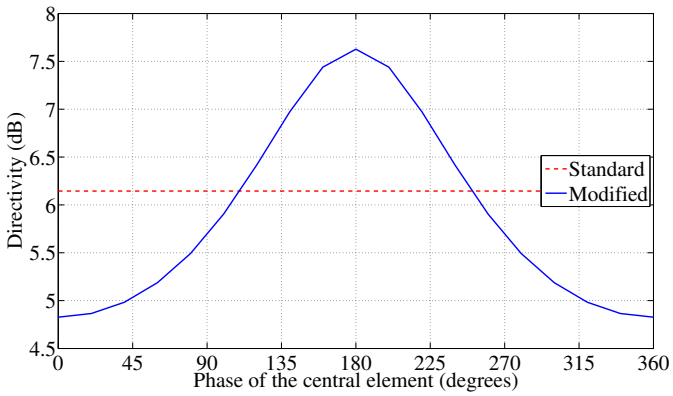


Figure 4.4: Directivity and phase of the central element.

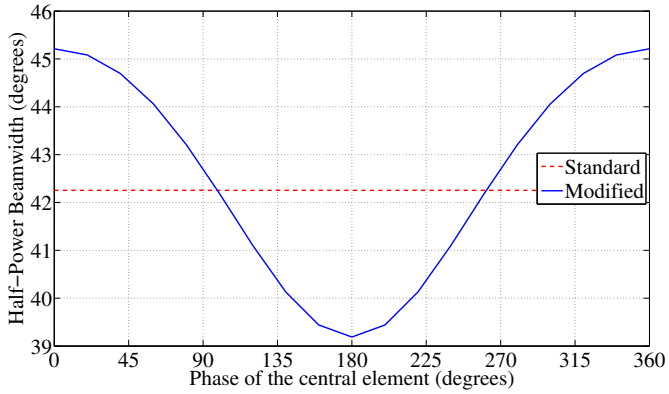


Figure 4.5: HPBW and phase of the central element.

Figure 4.6 shows an example for both antennas steering at $\theta = 20°$. It can be observed that the modified UCA achieves a higher directivity than the standard UCA at the desired angle.

Results in terms of side-lobe level were also obtained. A polar plot of relative directivity is shown in Figure 4.7, where the modified UCA exhibits a lower side-lobe level compared with the standard UCA.

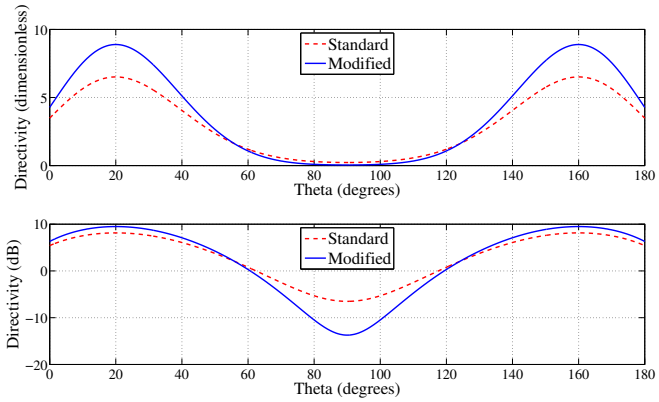


Figure 4.6: Directivity in dB and dimensionless.

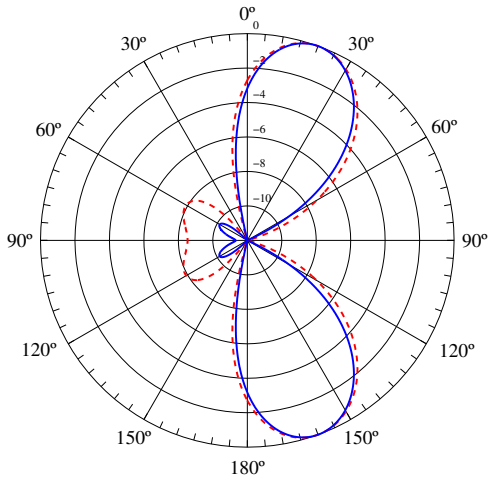


Figure 4.7: Polar plot of relative directivity.

4.4.3 Experiment 3: Design with Microstripes tool

Besides Matlab simulations, a 3D electromagnetic simulation tool called Microstripes [70] was used to plot the radiation pattern of the standard and modified UCAs. Microstripes is used extensively for solving challenging radiation problems including complex antenna structures. Two antenna array designs with 6 elements each were simulated to compare the standard and modified architectures. The results show a decrease of the side-lobe levels in the modified array. See Figures 4.8 and 4.9.

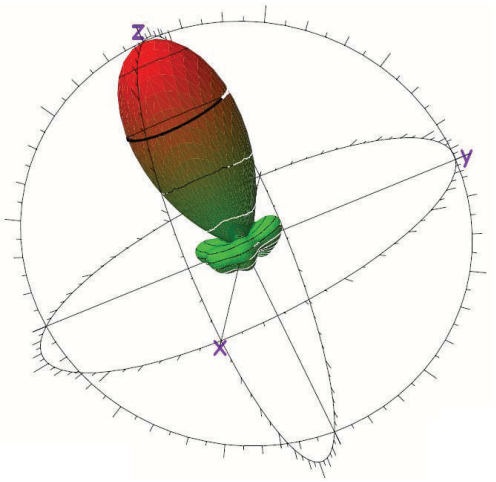


Figure 4.8: 3D standard array pattern.

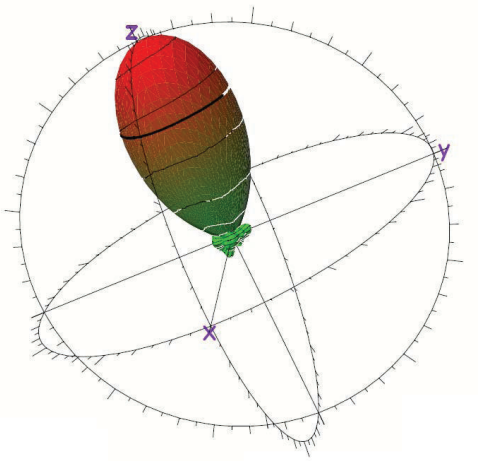


Figure 4.9: 3D modified array pattern.

The Microstripes simulations were performed for a variety of frequencies in which the antenna can transmit. Figure 4.10 shows the set of frequencies and the associated directivity. It can be seen that for almost all the frequencies, the directivity is higher for the modified UCA compared with the standard design.

The radiation pattern for both antennas was obtained from Microstripes. In Figure 4.11(a), it can be seen that the modified array presents lower values of side-lobe levels. Figure 4.11(b) is a polar plot showing the same data.

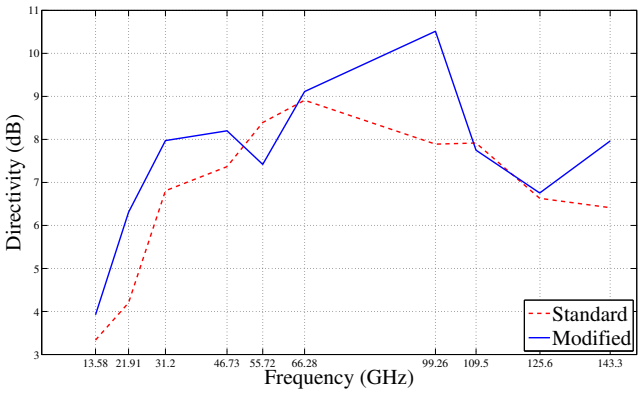
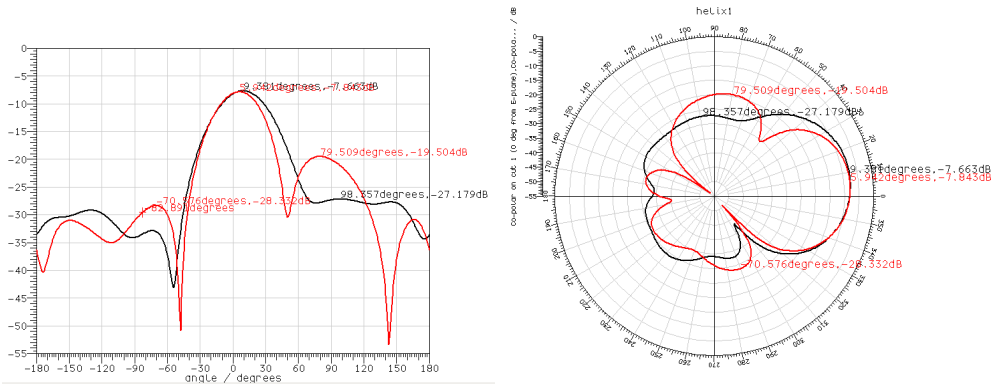


Figure 4.10: Directivity against frequency.



(a) Rectangular field pattern plot.

(b) Polar field pattern plot.

Figure 4.11: Radiation pattern for standard (red) and modified (black) circular arrays.

4.5 Non isotropic elements

In the previous subsection, the antenna elements used to obtain Directivity and HPBW measurements were considered isotropic. In an isotropic element, the directivity equals to 1 since the radiation occurs in all directions. But in a real world application, the antenna elements have a specific structure which affects the interference between the elements of an antenna array, thus changing the resulting radiation pattern. Moreover, the particular design of the entire array influences the antenna response to different transmission frequencies. This should also be taken into account when designing antenna arrays for real applications. For these reasons, the present work also includes the following tests of the modified uniform circular antenna array. The experiments consist in designing an UCA using Microstripes and simulations are performed for different ranges of frequencies to obtain data that allows a comparison between the standard and the modified UCA.

To perform these experiments, a real application circular antenna array design was used. This design, amongst others, is part of the work carried out by the ESPACENET project [71] which targets the development of flexible and intelligent embedded networked systems for aerospace applications. The aim is to develop a network architecture which can be applied to a constellation of micro satellites, which would one day replace existing large multifunctional satellites. The MEMS antennas designed by the group are constructed on the top substrate and have through wafer vias connecting the antennas to the MEMS and control structures.

4.5.1 Operation Frequencies

A series of simulations were performed using Microstripes for the proposed Modified Circular Array using the research group's design of the antenna element. The array consists of a ring of 8 elements and a ninth in the centre. Figure 4.12 a picture of the array.

Results for 10 different frequencies were obtained: 13.58, 21.91, 31.2, 46.63, 55.73, 66.28, 92.26, 109.49, 125.55 and 143.32 GHz. Table 4.1 summarizes the

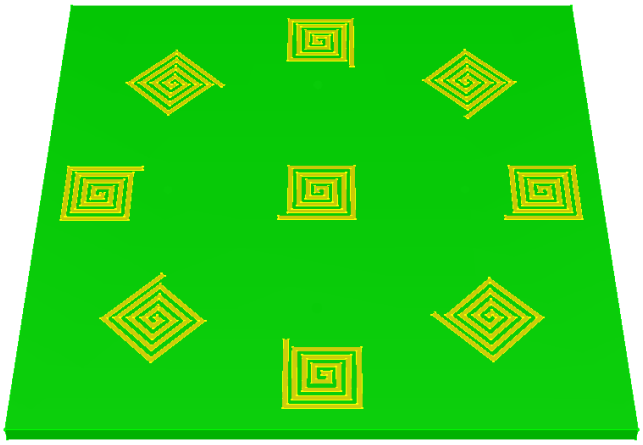


Figure 4.12: Modified Circular Array using the antenna elements proposed by the research group.

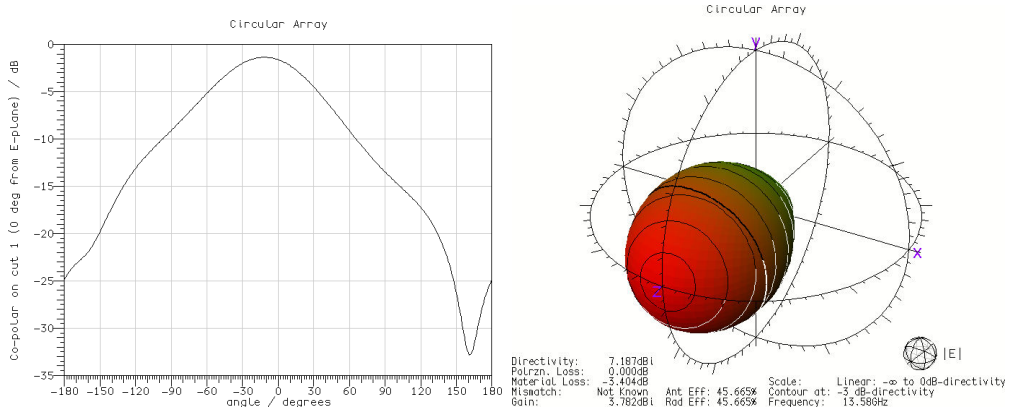
Table 4.1: Antenna results over a different range of frequencies.

Frequency (GHz)	Directivity (dBi)	Gain (dBi)	Antenna efficiency (%)	Material loss (dB)
13.58	7.18	3.78	45.66	-3.40
21.91	10.67	8.50	60.60	-2.17
31.20	13.50	12.33	76.40	-1.16
46.63	14.56	13.49	78.16	-1.07
55.73	13.65	13.22	90.53	-0.43
66.28	12.86	12.46	91.39	-0.39
92.26	11.39	10.57	82.70	-0.82
109.49	9.17	7.96	75.61	-1.21
125.55	8.65	7.12	70.15	-1.53
143.32	10.30	9.11	76.16	-1.18

results for bandwidth, directivity, gain and efficiency of the antenna at the respective frequencies. In terms of directivity, it can be seen that the levels are low for the low frequency of 13.58GHz, but it increases with frequencies 21.91GHz and 31.20GHz. The directivity is at its highest (14.56dBi) when the frequency value is 46.63GHz and then it decreases as the frequency augments from 55.73GHz to 125.55GHz. It is worth notice that for the frequency of 143.32GHz the directivity increases again, this time to 10.3dBi. A similar behaviour can be found for the

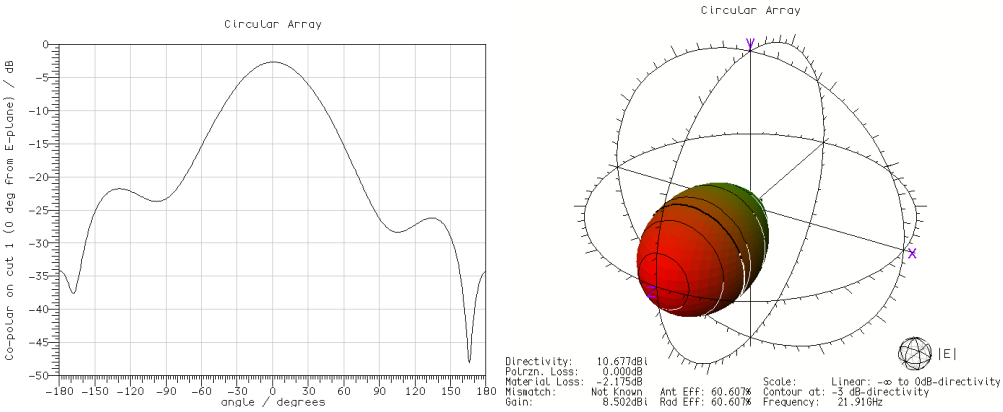
gain, which maximum value of 13.49dBi occurs when the frequency is 46.63GHz. The antenna efficiency follows the same pattern as the directivity, with the highest value of 90.53% when the frequency is 55.73GHz. Also the material loss, measured in dB, is lower with the lower frequencies of 13.58GHz and 21.91GHz, and its highest value of -0.39 occurs with the frequency of 66.28GHz.

Figures 4.13 to 4.22 show graphics obtained with Microsripes for each frequency respectively. On each figure, a 2-dimensional plot of the array pattern is shown as well as a 3-dimensional representation of the main lobe. Figure 4.13 shows a lobe with almost no sidelobe levels for frequency 13.58GHz, the same can be noticed in Figure 4.14 for the frequency of 21.91GHz. Figures 4.15 and 4.16 show the array pattern for frequencies 31.20GHz and 46.63GHz, this time it can be seen that there are small sidelobes, although the directivity is at its highest for these frequencies as discussed previously in Table 4.1. In the case of Figures 4.17 and 4.18 the sidelobes begin to show getting closer to the main lobe and the directivity decreases. Finally, Figures 4.19 to 4.22 show patterns where the main lobe is lost among the sidelobes which suggest that the antenna design being tested (Figure 4.12) is unable to operate under the frequencies from 92.26GHz to 125.55GHz.



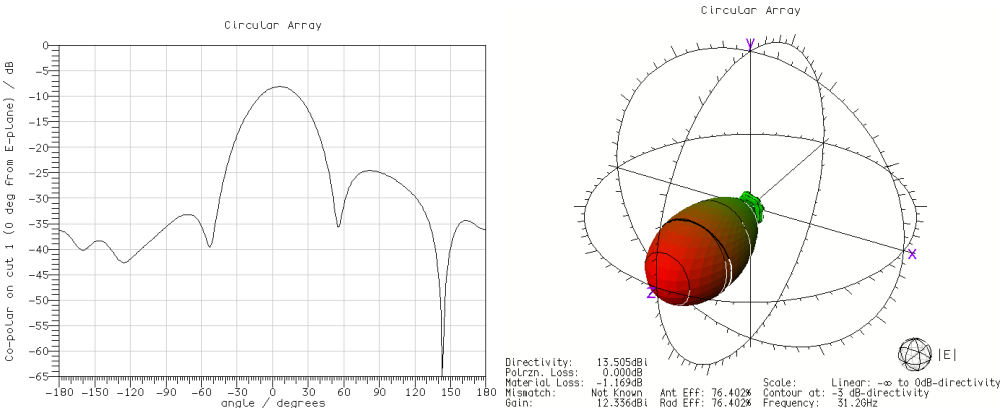
(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.13: Frequency 13.58GHz. Directivity: 7.187dBi.



(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.14: Frequency 21.91GHz. Directivity: 10.677dBi

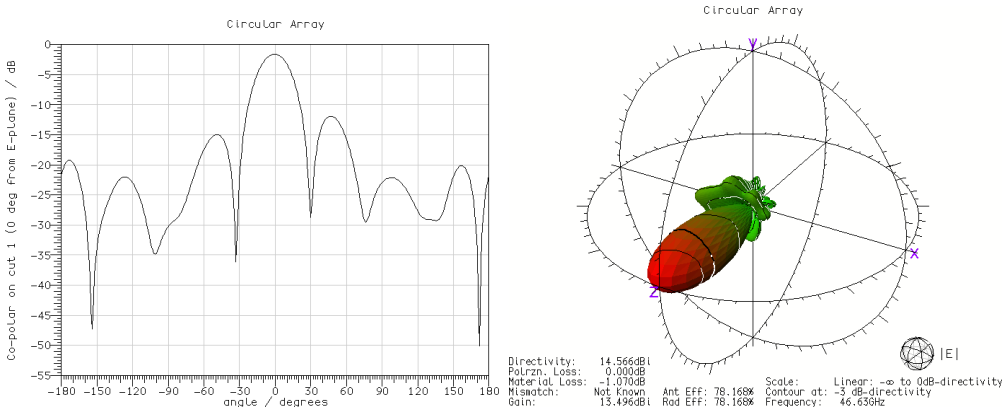


(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.15: Frequency 31.20GHz. Directivity: 13.505dBi

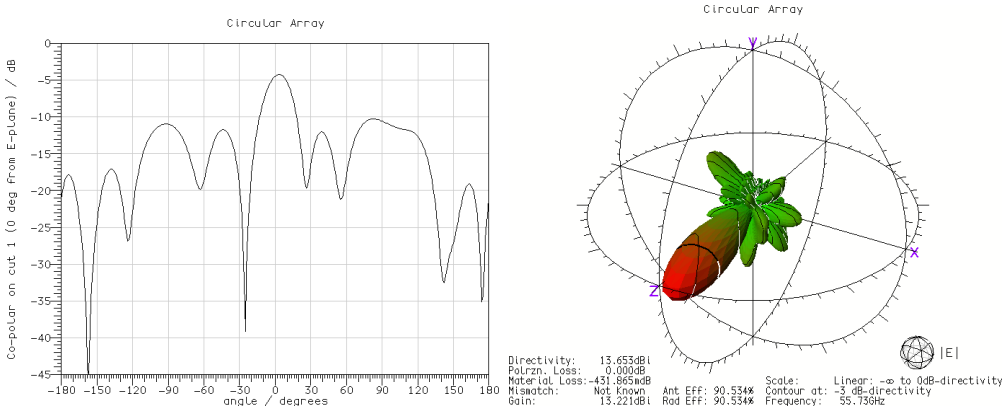
4.6 Summary

A geometry modification to the conventional uniform circular antenna array has been proposed. This modification consists in the placement of one of the antenna elements at the centre of the array. This element, modifies the overall radiation pattern in such a way that the directivity is increased whilst the half-power beamwidth angle is reduced. The result is a better capability of transmission in the desired direction and avoiding unwanted signals. It was also observed that the sidelobe levels of the radiation pattern were lower than those of the conventional circular antenna array which also helps to avoid interference. It



(a) 2-dimensional plot. (b) 3-dimensional plot.

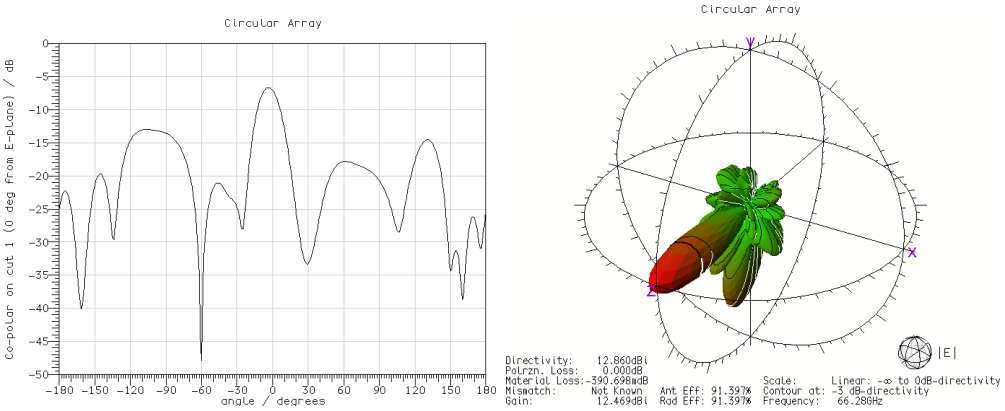
Figure 4.16: Frequency 46.63GHz. Directivity: 14.566dBi



(a) 2-dimensional plot. (b) 3-dimensional plot.

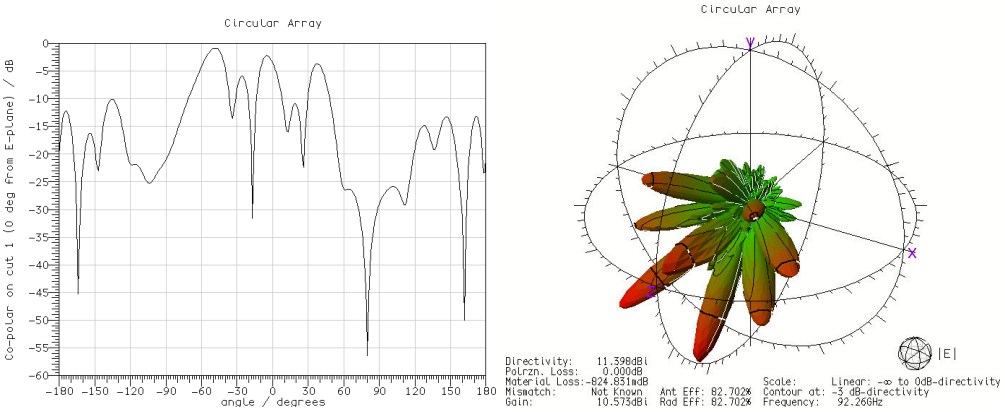
Figure 4.17: Frequency 55.73GHz. Directivity: 13.653dBi

was concluded that, to obtain the best trade-off, the circular array should be conformed by 6 antenna elements, which is the configuration that shows better directivity and reduced half-power beamwidth.



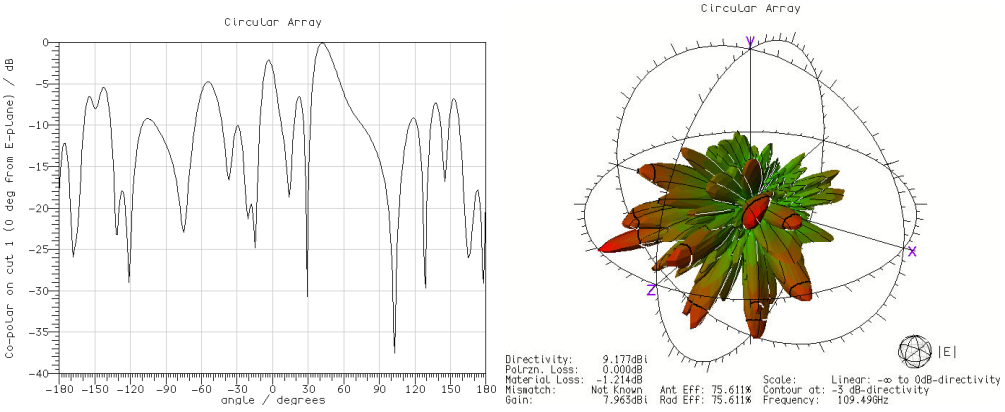
(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.18: Frequency 66.28GHz. Directivity: 12.860dBi



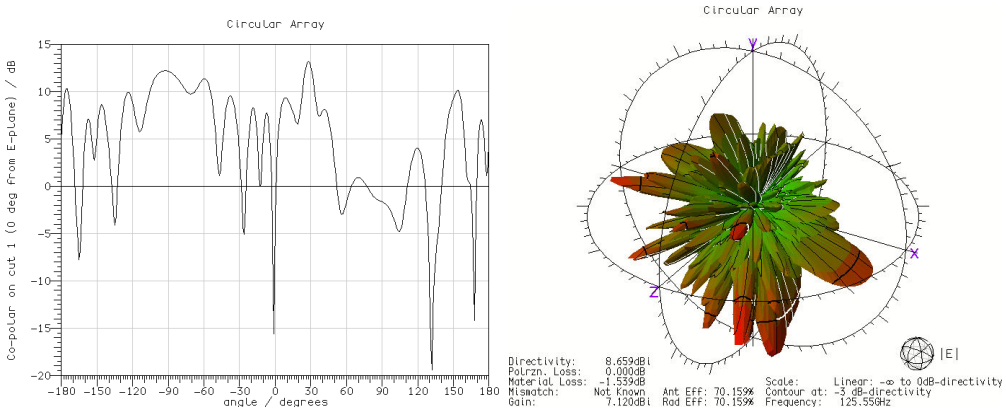
(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.19: Frequency 92.26GHz. Directivity: 11.398dBi



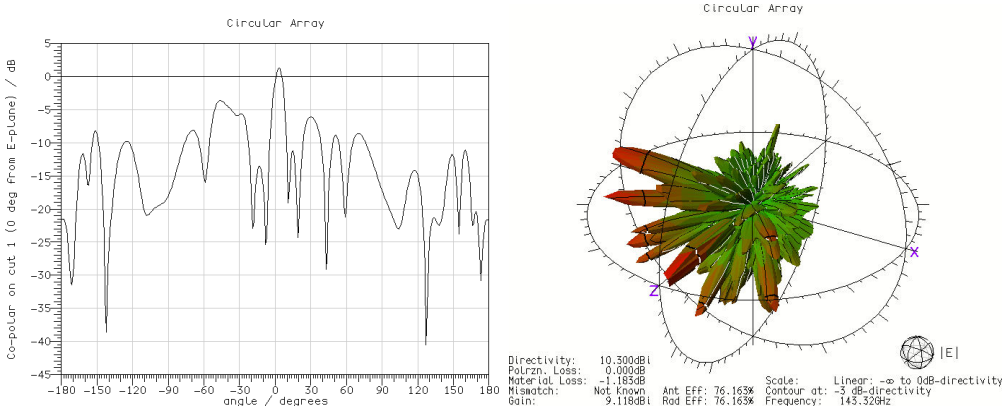
(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.20: Frequency 109.49GHz. Directivity: 9.177dBi



(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.21: Frequency 125.55GHz. Directivity: 8.659dBi



(a) 2-dimensional plot. (b) 3-dimensional plot.

Figure 4.22: Frequency 143.32GHz. Directivity: 10.300dBi

Chapter 5

Bio-inspired Algorithms for Radiation Pattern Optimization

In this chapter, an optimal radiation pattern is obtained for a linear antenna array using the particle swarm optimization technique. A set of phase shift weights is generated in order to steer the beam towards any desired direction while keeping nulls in the direction of interferers. The fitness function which allows the calculations of the phase shift weights is presented. A comparison between the standard genetic algorithm and the particle swarm optimization was studied and the results show that the latter achieves a better and more consistent radiation pattern than the GA. Moreover, a number of experiments show that the PSO is capable of solving the problem using less number of fitness function evaluations on average.

5.1 Introduction

Wireless communication technologies have experienced a fast growth in recent years. The latest mobile devices offer multi-bandwidth services and to enable this, new technologies have to be developed. Spatial processing is considered the last frontier in the battle for improved cellular systems and smart antennas are emerging as the enabling technique. The use of adaptive antenna arrays in mobile handsets can help eliminate co-channel interference and multi-access

interference amongst other problems. These breed of antennas are able to radiate power towards a desired angular sector, thus, avoiding interference with undesired devices. Figure 5.1 shows a graphic explanation of this concept. The number, geometrical arrangement, and relative amplitude and phases of the array elements depend on the angular pattern that must be achieved. By changing the relative phases of array elements, a process called steering, an array is capable of focusing its main beam towards a particular direction.

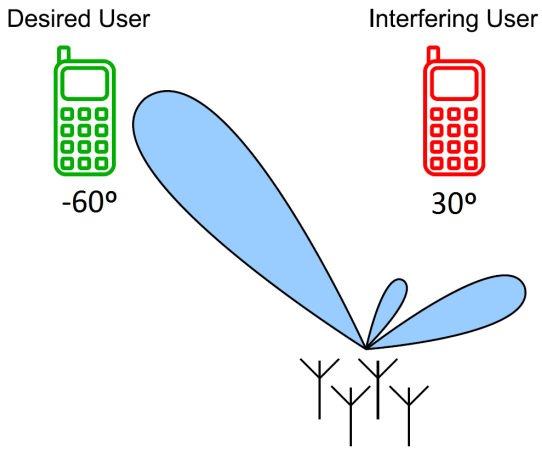


Figure 5.1: Radiation pattern for desired and interfering angles -60°, 30°.

This thesis is part of an ongoing research at the System Level Integration research group [72, 73, 74]. This research focuses on reconfigurable MEMS (Micro-Electro-Mechanical Systems) sensors and antennas. The group has developed a series of phased array antennas which are ideal for reconfigurable networks. MEMS devices offer very low loss switching which means that the RF network of MEMS switches will not interfere or degrade antenna radiation patterns. MEMS can be used in several ways to achieve reconfigurability. One of these options is to employ MEMS switches to connect the antenna elements. These phase shifters are designed to alter the phase of the signal on the transmission line. They are fabricated from the same material as the antenna and much smaller than DMTL (Distributed MEMS Transmission Line) phase shifters. This approach allows the design of smart antennas capable of steering its radiation pattern towards a given direction by configuring the relative phases of the array elements. Figure 5.2 shows a diagram of a general phase-shift smart antenna system.

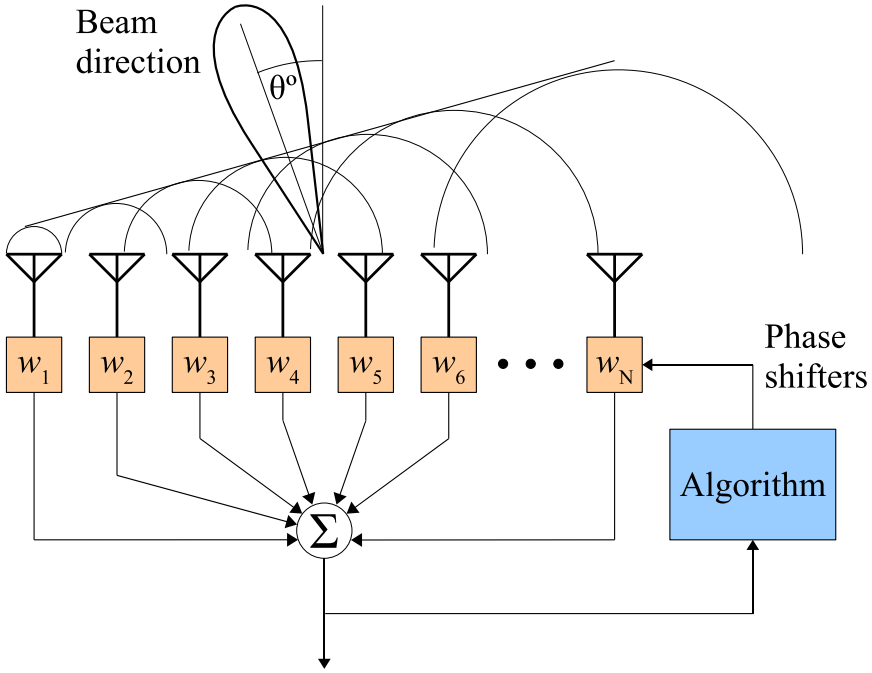


Figure 5.2: Diagram of a Phase-shift Smart Antenna system.

Although, as noted in Chapter 3, for a beamsteered array to be a smart antenna it has to use signal processing in order to obtain the desired beam pattern according to certain conditions. Thus, an algorithm that controls the antenna performance is needed. In the case of phase-shift arrays, the phase of each antenna element must be set by a computer control or intelligence.

5.2 Adaptive Antenna Arrays and Bio-Inspired Algorithms

Due to the amazing development of computers, the application of numerical optimization techniques to antenna design has become possible. Genetic algorithms have been applied to adapting the response of an antenna array in order to reject interference. In [7], a constrained GA was used to prevent nulling of the desired signal received by the main beam. This work used amplitude and phase, phase only and amplitude only weights. Wang et al. [8] also proposed the optimization of amplitude and phase with genetic algorithms using a combined approach

including subarray amplitude weights. Their results show that the combined approach reduces the grating lobes successfully. In [9], the Bees Algorithm (BA) was used for the pattern synthesis of a linear antenna array with prescribed nulls. Nulling of the pattern is achieved by controlling the amplitude of each array element. On the other hand, a phase-only technique was proposed in [75] where local genetic algorithms were used to search for the optimal weighting vector of the phase shift perturbation of the antenna array. A new convergent method referred to as the two-way convergent method was presented.

Other bio-inspired algorithms like the Particle Swarm Optimization (PSO) [10] have been found to be effective in optimizing difficult multidimensional problems in a variety of fields [76] including electromagnetics [77, 78]. This technique has proven to be successful for antenna design, as presented in [79, 15, 78] and has been shown to outperform, in certain cases, other optimization methods [80]. The PSO algorithm has been used to reconfigure phase-differentiated array antennas in [11]. In this work, element excitations are found that produce a main beam with low sidelobes with the additional requirement that the same excitation amplitudes should result in a high directivity and pencil-shaped main beam.

As described in Chapter 3, Particle Swarm Optimization is based on the behaviour of groups of living creatures like a swarm of bees. Their goal is to find the location with the highest density of flowers by randomly flying over the field. Each bee can remember the location where it found the most flowers, and by dancing in the air, it can communicate this information to other bees. Occasionally, one bee may fly over a place with more flowers than had been previously discovered by any bee in the swarm. Over time, more bees end up flying closer and closer to the best patch of the field. Soon, all the bees swarm around this point.

The suitability of the PSO algorithm for adaptive arrays has been described in [81], where it was employed for blind adaptation of the directional characteristic of antenna arrays. It was found that the PSO is capable of following the dynamic changes in the environment and the possibility of an FPGA implementation was discussed. Modifications to the PSO algorithms have been presented by Li et al. [82], where the EPSO (Extended Particle Swarm Optimization) was proposed.

This extension uses a velocity updating mechanism and new exceeding boundary control operators to overcome the drawbacks of PSO.

Previous work on the field of antenna array analysis and design using PSO have been presented in [15], where the relative position of the antenna elements has been optimized to obtain minimum Side-Lobe Levels (SLL) and nulls towards the undesired directions. The PSO algorithm has successfully been also applied to design other kinds of antennas like circular antenna arrays [83] by setting the distance between the elements. However, in the case of smart antennas, the position of the antenna elements is fixed so the relative displacement can not be changed. To determine the shape of the radiation pattern, another characteristic of the array must be adjusted, for example the excitation phase of each individual element. Phase shifters connected to the antennas can be used to cancel interference by placing nulls on the directions of the interfering sources. This was proposed in [84] and was accomplished by using Memetic Algorithms.

In this chapter, the use of the Particle Swarm Optimization technique to find the optimal radiation pattern of an adaptive antenna is proposed. By calculating the phase shift weights of a linear antenna array, the beam direction can be steered towards a desired angle. In addition, it is possible to place nulls at the direction of possible interferers. The present work compares these results with the ones obtained by using a Genetic Algorithm (GA) and shows that the PSO performs better in terms of power levels. Furthermore, for a desired array configuration, the number of fitness function evaluations performed by the PSO is shown to be less than the one from the GA. This can lead to an improvement in the overall performance of an adaptive array since its configuration must meet tough demands.

5.3 Problem Description

Let us assume that the antenna under investigation is an array of $2N$ infinitesimal dipoles positioned along the x-axis equidistant from each other as shown in Figure 5.3. The total field of the array is equal to the field of a single element positioned at the origin multiplied by a factor which is widely referred to as the *array factor*.

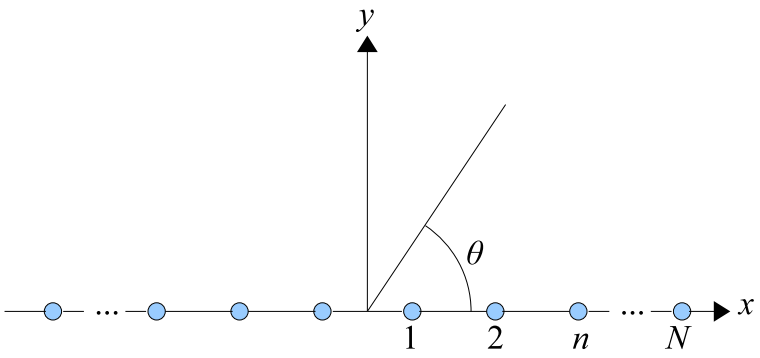


Figure 5.3: Linear array of 2N elements positioned along the x-axis.

Thus, for the $2N$-element array, the array factor is given by [23]

$$AF(\theta) = \sum_{n=1}^{2N} w_n e^{j(n-1)(kd\ cos(\theta)+\beta)} \tag{5.1}$$

$\psi = kd\ cos(\theta) + \beta$

If the distance among elements is d and the reference point is the centre of the array, the array factor becomes

$$AF(\theta) = \sum_{n=1}^{2N} \alpha_n e^{j[(n-N-0.5)\psi+\beta_n]} \tag{5.2}$$

where

$2N$ = number of antenna elements

α_n = amplitude weight at element n

β_n = phase shift weight at element n

$\psi = \frac{2\pi}{\lambda} d \sin(\theta) = kd \sin(\theta)$

θ = angle of interfering or desired signal

In this work, only the phase shift weights are considered, so the amplitude weights are constant. If the phase shifts are odd symmetry, the array factor can be written as

$$AF(\theta) = 2\sum_{n=1}^{N} \cos[(n-0.5)\psi + \beta_n] \tag{5.3}$$

and in its normalized form

$$AF_n(\theta) = \frac{1}{N}\sum_{n=1}^{N} \cos[(n-0.5)\psi + \beta_n] \tag{5.4}$$

This equation represents a mathematical description of the antenna radiation pattern and can be used by optimization algorithms. The PSO algorithm is able to search for optimal phase shift weights using a fitness function based on this array factor.

5.4 Simulations

This section presents the three main experiments performed to demonstrate the performance of the Particle Swarm Optimization and the Genetic Algorithm. The first experiment consists in simulating a 20-element linear antenna array with 1 desired transmitter and 1 interferer. Both the PSO and the GA algorithms will obtain an optimum phase-shift vector that configures the array in such a way that the resulting main lobe points towards the desired signal, while the array pattern presents a null in the direction of the interferer. The aim of the experiment is to compare the results obtained by the algorithms in terms of average number of function evaluations. As mentioned before, this measure describes the capability of an algorithm to find an acceptable solution and is an important factor in the performance of mobile devices. The second experiment is similar to the first one and the aim is to test both algorithms to a more difficult environment, namely having two undesired signals instead of one. Given the difficulty of obtaining a suitable array pattern, the second experiment is performed for a 40-element linear antenna array. Lastly, a third experiment is run to investigate the possibility of having two desired signals instead of one. Once again, the PSO and the GA algorithms are tested for a 40-element antenna.

5.4.1 Experiment 1: 1 desired and 1 undesired signals

Given a desired transmitter called $user1$ at the direction of -60° and an interferer transmitter called $user2$ at 30°, find a set of phase shifters that will configure a linear antenna array in such a way that the main lobe is directed to $user1$ whilst a null is presented to $user2$. For this problem, an array of 20 isotropic elements is defined, so $N = 10$ which is the dimension of the problem and the result consists of a vector x of 10 elements, each one corresponding to β_n. Figure 5.1 shows an illustration of the desired radiation pattern.

The geometry of the linear array is defined as follows: The distance d of any two adjacent elements is set to $\lambda/2 = 2$ where λ is the wavelength. k equals to $(2\pi)/\lambda$ and represents the wavenumber. The PSO algorithm is programmed in Matlab and is based on the Standard PSO 2007 proposed by Maurice Clerc in [85]. The parameters are set as shown in Table 5.1.

Table 5.1: Set of parameters for the PSO.

Swarm size	20
Inertia	1.0
Correction factor	2.0
Minimum boundary	$-\pi$
Maximum boundary	π
Tolerance	1x10^{-6}

The swarm size is set to 20 individuals. The inertia corresponds to the weight w and is fixed to 1.0. The correction factor are the constants c_1 and c_2 in the PSO velocity Equation 3.1. Previous work has shown that a value of 2.0 is a good choice for both parameters [86]. Initial positions are chosen at random inside the search space which is $[-\pi; \pi]$ radians. This means that the algorithm must be configured to limit each particle position to those constraints. If a particle moves out of the the search space, its position is set to the previous value. The algorithm stops when the difference between successive fitness function values is less than the tolerance value, in this case, 1x10^{-6}. Given that there are two conditions to be met, the fitness function consists of two parts: $F(\theta_1)$ which will attempt to maximize the value of the array factor for the direction of $user1$ = -60°. While a

second function, $F(\theta_2)$ must minimize the array factor for the direction of $user2$ = 30°. The following fitness function is deduced

$$FitnessFunction = F_1 - F_2 \tag{5.5}$$

where

$$\begin{aligned} F_1 &= |AF(\theta_1)|^2 \\ &= \left| \frac{1}{N} \sum_{n=1}^{N} \cos[(d-0.5)k\sin(\theta_1) + \beta_n] \right|^2 \end{aligned} \tag{5.6}$$

and

$$\begin{aligned} F_2 &= |AF(\theta_2)|^2 \\ &= \left| \frac{1}{N} \sum_{n=1}^{N} \cos[(d-0.5)k\sin(\theta_2) + \beta_n] \right|^2 \end{aligned} \tag{5.7}$$

where θ_1 and θ_2 correspond to the angles of $user1$ and $user2$ respectively.

A standard Genetic Algorithm (GA) is tested to compare its performance against the PSO. The Matlab function *ga* from the "Genetic Algorithm and Direct Search Toolbox" is used. All parameters are left to their default values (see Table 5.2 except for the population size which is set to 20, the maximum number of generations which is 500, the defined boundaries $-\pi$ and π and the tolerance value which is the same as the PSO: 1x10^{-6}. Using the same tolerance for both algorithms ensures that they will attempt to reach a result within the same error of each other.

After running the simulations, a vector of phase shift weights for each algorithm is obtained, x_{ga} and x_{pso} as shown in Table 5.3.

The algorithm initializes each particle with a set of randomly generated shift weights. These values are optimized at each iteration until their values produce an array pattern that meets the given constraints. This optimization is shown as a change in the values of the phase-shift vector which is represented by the position of each particle. Figure 5.4 is a graph that shows the behaviour of one of the particles for this run. It can be seen that the particle's position converges

Table 5.2: Set of parameters for the GA.

Population size	20
Individual's encoding	Real
Selection	Stochastic uniform
Reproduction	Elite count: 2. Crossover fraction: 0.8
Mutation	Constraint dependent default
Crossover	Scattered
Migration	Forward. Fraction:0.2. Interval: 20

Table 5.3: Phase shift vectors (in radians) to obtain a main lobe at -60°, null at 30°.

x_{ga}	x_{pso}
$\beta_1 = 1.4378$	$\beta_1 = 1.2899$
$\beta_2 = -0.7943$	$\beta_2 = -2.2126$
$\beta_3 = 1.0848$	$\beta_3 = 0.5883$
$\beta_4 = 2.6372$	$\beta_4 = 3.0162$
$\beta_5 = -0.2880$	$\beta_5 = -0.3530$
$\beta_6 = 2.3845$	$\beta_6 = 2.4889$
$\beta_7 = -0.1420$	$\beta_7 = -1.2133$
$\beta_8 = 1.9148$	$\beta_8 = 1.5080$
$\beta_9 = -0.8593$	$\beta_9 = -1.9363$
$\beta_{10} = 0.8625$	$\beta_{10} = 0.7529$

to the optimum values on each iteration until the stop criteria is met.

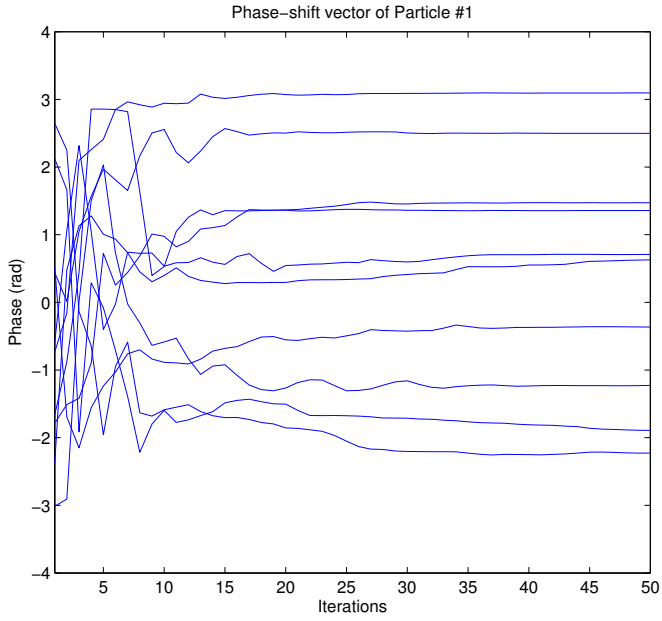


Figure 5.4: Phase-shift vector of one of the particles for a 20-element array.

The resulting radiation pattern in dB (decibels) is shown in Figure 5.5. As can be seen, both algorithms successfully obtained a suitable set of phase shift weights that produce the desired radiation pattern. The main lobe is directed towards the angle of $user1$ = -60°, while a null in the direction of $user2 = 30°$ is formed. It can also be noted that the PSO algorithm achieved a better radiation pattern than that obtained by the GA. The power of the main lobe is also higher for the PSO as the value of the null is lower. It was also observed that the PSO algorithm performed a lower number of fitness function evaluations than the GA. The PSO executed 1338 fitness function operations whereas the GA executed 1667.

To further study this, the same experiment was run 100 times and the mean was calculated. Another factor to consider is the dimension of the problem or number of array elements. In Figure 5.6, the graph shows the total number of function evaluations executed by both algorithms over different dimensions. For a range of dimensions from 5 to 30, the GA performed more function evaluations than the PSO. This is important since a dynamic configuration of the array

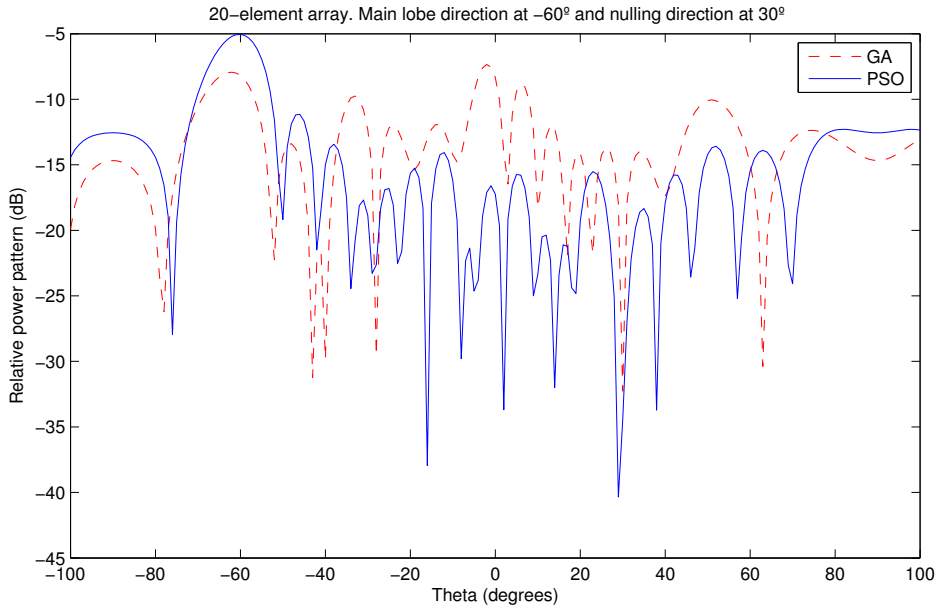


Figure 5.5: Radiation pattern for desired and interfering angles -60°, 30°.

should meet the performance requirements according to the application. The total number of fitness function evaluations indicates the overall performance of the system, given that the fitness function evaluation is the most computationally intensive part of the algorithm.

5.4.2 Experiment 2: 1 desired and 2 undesired signals

A second experiment was performed in which the capability of avoiding two different directions instead of one was tested. In this example, $user2$ is moved to the direction -20°, closer to the desired $user1$ which remains at -60°. A third undesired user, $user3$ appears in the direction 40°. The aim is to create nulls in $user2$ and $user3$ directions. This time, the number of antenna elements is set to 40 so the dimension of the problem is $N = 20$. The rest of the parameters for both algorithms are the same as the first experiment. A set of 20 phase shift weights is obtained from the GA and PSO algorithms as shown in Table 5.4.

Figure 5.7 show the convergence of the phase-shift vector of one of the particles. It can be seen that the optimum values are obtained around the 70^{th} iteration.

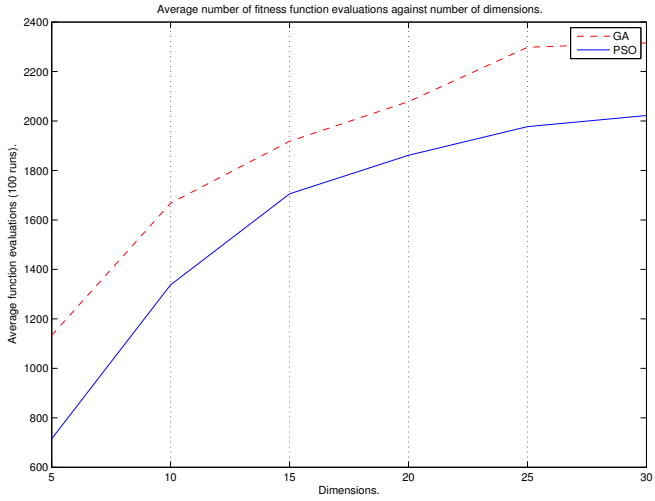


Figure 5.6: Average number of function evaluations against dimensions.

Table 5.4: Phase shift vectors for main lobe at -60°, nulls at -20°, 40°.

x_{ga}	x_{pso}
$\beta_1 = 1.7244$	$\beta_1 = 1.5439$
$\beta_2 = 0.6048$	$\beta_2 = -2.1608$
$\beta_3 = 0.6785$	$\beta_3 = 0.4752$
$\beta_4 = 2.3043$	$\beta_4 = 3.1413$
$\beta_5 = 0.6309$	$\beta_5 = -0.2736$
$\beta_6 = 1.9293$	$\beta_6 = -3.1392$
$\beta_7 = 0.9169$	$\beta_7 = -1.2152$
$\beta_8 = 1.2024$	$\beta_8 = 1.2778$
$\beta_9 = 2.2653$	$\beta_9 = 3.1415$
$\beta_{10} = 0.9047$	$\beta_{10} = 0.7004$
$\beta_{11} = 3.1094$	$\beta_{11} = -2.9015$
$\beta_{12} = 0.9034$	$\beta_{12} = -0.0983$
$\beta_{13} = 2.4115$	$\beta_{13} = -3.1415$
$\beta_{14} = 0.3607$	$\beta_{14} = -0.9601$
$\beta_{15} = 1.1969$	$\beta_{15} = 1.4342$
$\beta_{16} = 0.3128$	$\beta_{16} = -1.7718$
$\beta_{17} = 1.2371$	$\beta_{17} = 0.9627$
$\beta_{18} = 2.8380$	$\beta_{18} = -2.7192$
$\beta_{19} = 0.3142$	$\beta_{19} = 0.0213$
$\beta_{20} = 2.8321$	$\beta_{20} = 2.8388$

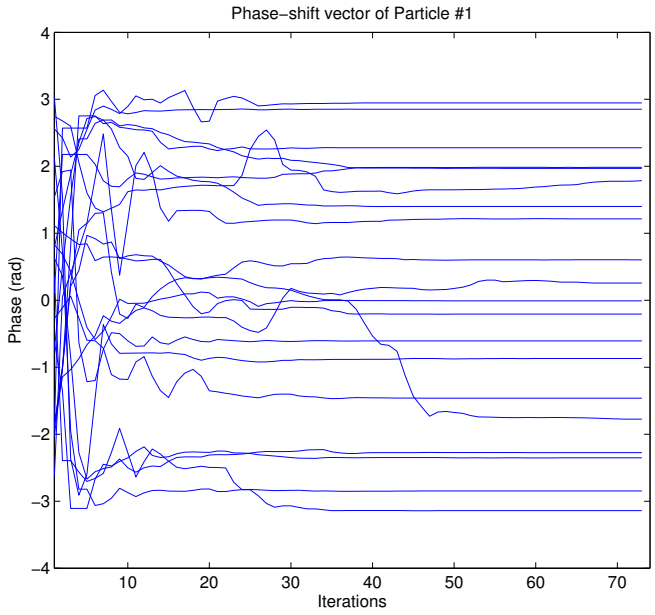


Figure 5.7: Phase-shift vector of one of the particles for a 40-element array.

It is important to note that the fitness function must be modified for this problem since there is a third factor that affects its output. The new fitness function is given by Equation 5.8.

$$Fitness = F_1 - F_2 - F_3 \tag{5.8}$$

where

$$\begin{aligned} F_1 &= |AF(\theta_1)|^2 \\ &= \left| \frac{1}{N} \sum_{n=1}^{N} \cos[(d-0.5)k\sin(\theta_1) + \beta_n] \right|^2 \end{aligned} \tag{5.9}$$

,

$$\begin{aligned} F_2 &= |AF(\theta_2)|^2 \\ &= \left| \frac{1}{N} \sum_{n=1}^{N} \cos[(d-0.5)k\sin(\theta_2) + \beta_n] \right|^2 \end{aligned} \tag{5.10}$$

and

$$\begin{aligned} F_3 &= |AF(\theta_3)|^2 \\ &= \left| \frac{1}{N} \sum_{n=1}^{N} \cos[(d-0.5)k\sin(\theta_3) + \beta_n] \right|^2 \end{aligned} \tag{5.11}$$

where F_3 corresponds to the third condition which is a null at the direction of $user3 = 40°$. After simulating the GA and PSO algorithms, the resulting radiation pattern, shown in Figure 5.8, is obtained.

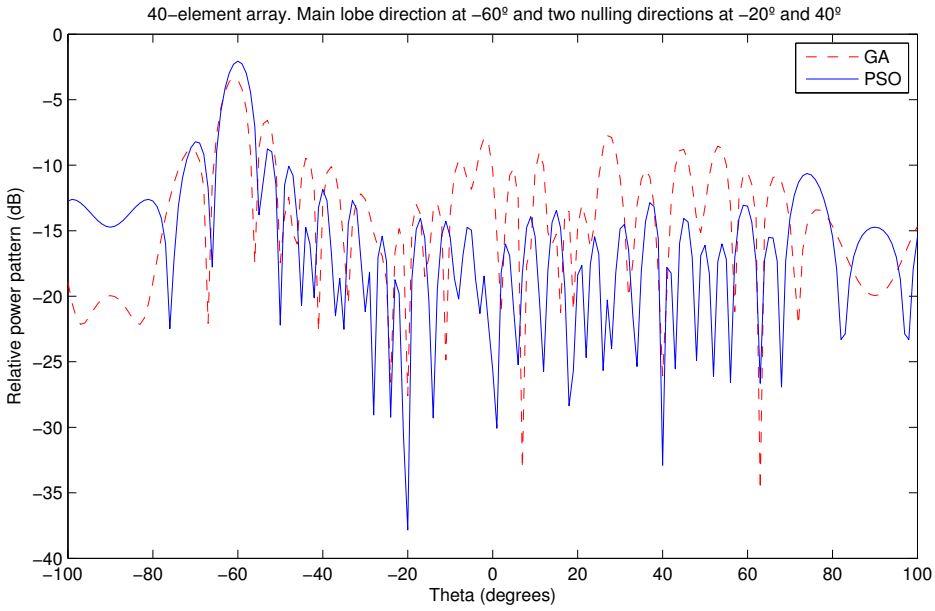


Figure 5.8: Radiation pattern for desired angle -60°, and interfering angles -20° and 40°.

Once again, it can be observed that both algorithms are able to optimize a set of phase shift weights that cause a main lobe and two nulls to point to the desired and interfering directions. Similar to the first experiment, the PSO algorithm performs generally better in terms of radiating power. Moreover, it can be noticed that the Side-Lobe Levels (SLL) which are the lobes other than the main lobe, are considerably lower for the PSO compared to those for the GA. This is often desirable as it helps to avoid other interfering signals at different directions other than the main lobe. These experiments suggest that the PSO

algorithm tends to perform better than the GA algorithm in terms of higher power in the desired direction and lower power in the interfering directions. This lead to carrying out a third experiment to further study this: In order to observe if the PSO obtains in general better configurations, the second experiment with the same conditions was repeated 1000 times. The power at the main lobe and null directions (maximum and minimum levels) of $user1$, $user2$ and $user3$ for both algorithms was measured. The results of both PSO an GA algorithms are shown in Figure 5.9.

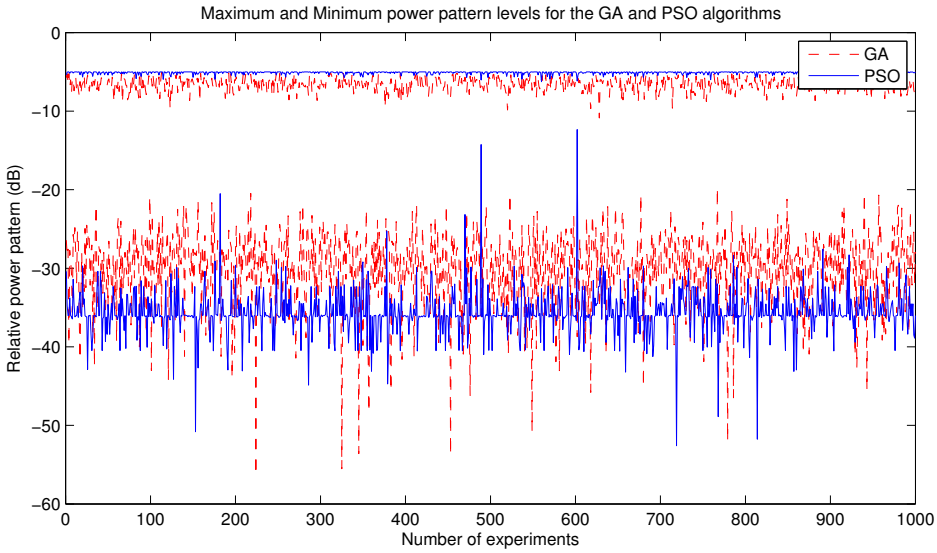


Figure 5.9: Main lobe and nulls power values for the GA and PSO algorithms.

It can be observed that the power of the main lobe obtained by the PSO algorithm is generally maintained around -5dB for almost all of the 1000 experiments while the levels achieved by the GA are not as constant and oscillate between -5dB and -10dB. Similarly, the power radiated towards the undesired directions is generally lower in the case of the PSO compared with the ones obtained by the GA.

5.4.3 Experiment 3: 2 desired and 3 undesired signals

A third experiment was conducted to investigate the possibility of having two desired users instead of one. This time, $user1$ and $user2$ have desired angles at

20° and 60° respectively. A number of three undesired directions is set at -70°, -50° and 0°. The resulting pattern of this experiment is shown in Figure 5.10.

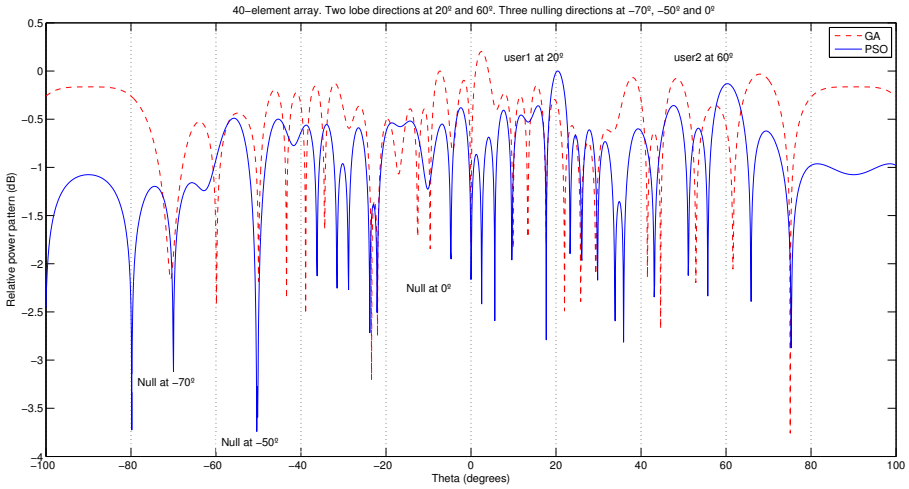


Figure 5.10: Radiation pattern for two desired angles at 20° and 60°, and interfering angles at -70°, -50° and 0°.

It can be seen that both algorithms managed to get prescribed nulls at the undesired directions, but the PSO was slightly better at directing the beams with the most power towards the desired users.

5.5 Summary

In this chapter, the Particle Swarm Optimization method was used to obtain a set of phase shift weights that configure a linear antenna array. These weights were optimized in order to maximize the power of the main lobe at a desired direction while keeping nulls towards interferers. A comparison with a Genetic Algorithm was studied and the results of 1000 experiments show that the PSO achieves better and more consistent radiation patterns than those of the GA. It was also observed that the total number of fitness function evaluations is lower for the PSO, which suggests an advantage in terms of performance as the function evaluation tends to have higher computational cost.

Chapter 6

Adaptive Antennas and Particle Swarm Optimization

In this chapter, the Particle Swarm Optimization algorithm is used to generate a set of array weights for a uniform planar rectangular array. The goal is to maximize the power towards a desired direction while minimizing it in the direction of interferers. A fitness function based on the Signal-to-Interference-plus-Noise Ratio is used. The results are compared with those obtained by the Genetic Algorithm. These results suggest that the PSO algorithm outperforms the GA in terms of the total number of fitness function evaluations. It was also concluded that the PSO is able to obtain lower sidelobe levels which help to avoid interference. Moreover, the gain levels are lower in the direction of interferers compared with those obtained by the GA.

6.1 Introduction

With the fast growth of mobile communication devices, new adaptive techniques are required to reduce the effects of interfering radiation. These methods must be able to conform to the ever changing environment conditions and maintain an efficient use of the communication channel. Adaptive antenna arrays are able to automatically extract the desired signal from interferer signals and external noise. Moreover, they are capable of continuously updating their array weights

to ensure the best quality of service is achieved [27]. This is attained by radiating power towards the desired angular sector, thus, excluding undesired signals from other incidence angles.

Several methodologies have been proposed in the past to find the optimal synthesis of array weights. One such technique was proposed by Applebaum in [87] where an algorithm for adaptive interference cancellation is presented. Together with the adaptive sidelobe cancellation (SLC) by Howells [88, 89] it became known as the *Howells-Applebaum algorithm*. Another approach has been proposed by Widrow et al. in [90] where by using the *least mean squares* (LMS) a self-training system for adaptive arrays was developed. Both the *Howells-Applebaum* and *Widrow* methods are gradient-search algorithms. Although these solutions are mathematically elegant, their implementation in hardware is very difficult due to the cost of the analog components. Some solutions require an expensive receiver at each antenna element. In order to make adaptive antennas commercially available, the hardware must use already available components which are in the digital domain. Digital phase shifters in standard array architectures turn out to be a cheap solution as the use of expensive adjustable amplitude weights or correlators is avoided.

Other approaches have been proposed where the synthesis is treated as an optimization problem using evolutionary techniques like the genetic algorithm. Genetic algorithms are ideal for solving such problems as they are capable of optimizing nonlinear multimodal functions that have many variables. Moreover, experimental results show that GAs are able to find good solutions to antenna array optimization in a quick manner [38]. In recent years, modern computer systems have enabled researchers to apply these techniques to antenna design [78, 11, 91]. Besides the GAs, another bio-inspired algorithm which is widely used is the PSO algorithm. It has been found to be effective in optimizing difficult multidimensional problems in a variety of fields [76, 78]. Unlike GAs, the PSO is based upon the cooperation amongst the individuals rather than their competition. In addition, it is easier to calibrate and to control the parameters of the PSO over the GA [39]. GAs require a specific strategy and careful choice of operators according to the application, whereas PSO eliminates the process of

selecting the best operators by sequentially updating its equations. Therefore, the aim of this work is to evaluate the effectiveness of the PSO in solving adaptive antenna array challenges and to analyse and compare its performance with other optimization techniques.

6.2 Mathematical Formulation

Let us consider a traditional narrowband antenna array of N elements as shown in Figure 6.1.

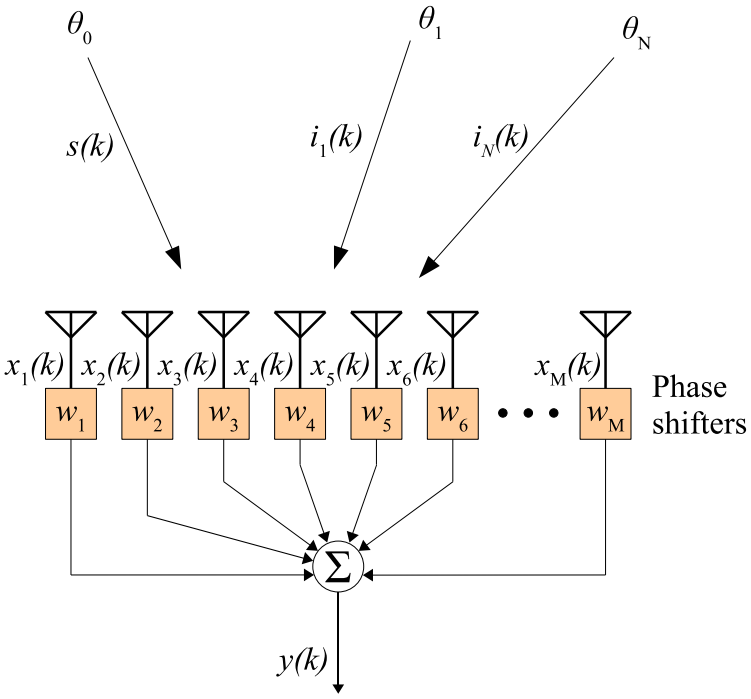


Figure 6.1: Traditional narrowband antenna array.

$s(k)$ is a desired signal arriving from the angle θ_0. $i_1(k)$ to $i_N(k)$ are N interferers arriving from angles θ_1 to θ_N. M elements with M phase weights receive the total signal composed by the desired signal and the interferers. The general total array output is given by [16]

$$y(k) = \bar{\omega}^H \cdot \bar{x}(k) \tag{6.1}$$

where

$$\begin{aligned} \bar{x}(k) &= \bar{a_0}s(k) + [\bar{a_1}\ \bar{a_2}\ \cdots\ \bar{a_N}] \cdot \begin{bmatrix} i_1(k) \\ i_2(k) \\ \vdots \\ i_N(k) \end{bmatrix} + \bar{n}(k) \\ &= \bar{x_s}(k) + \bar{x_i}(k) + \bar{n}(k) \end{aligned} \tag{6.2}$$

with

$\bar{\omega}$ = $[\omega_1\ \omega_2\ \cdots\ \omega_M]^T$ = array weights
$\bar{x_s}(k)$ = desired signal vector
$\bar{x_i}(k)$ = interfering signals vector
$\bar{n}(k)$ = zero mean Gaussian noise for each channel
$\bar{a_i}$ = M-element steering vector for the θ_i direction of arrival

The weighted array output power for the desired signal is

$$P_{ss} = E\left[\left|\bar{\omega}^H \cdot \bar{x_s}\right|^2\right] = \bar{\omega}^H \cdot \bar{R_{ss}} \cdot \bar{\omega} \tag{6.3}$$

where

$$\bar{R_{ss}} = E\left[\bar{x_s}\bar{x_s}^H\right] \tag{6.4}$$

is the signal correlation matrix. And the weighted array output power for the undesired signals is given by

$$P_{uu} = E\left[\left|\bar{\omega}^H \cdot \bar{u}\right|^2\right] = \bar{\omega}^H \cdot \bar{R_{uu}} \cdot \bar{\omega} \tag{6.5}$$

where

$$\bar{R_{uu}} = \bar{R_{ii}} + \bar{R_{nn}} \tag{6.6}$$

with

$\bar{R_{ii}}$ = correlation matrix for interferers
$\bar{R_{nn}}$ = correlation matrix for noise

The Signal-to-Interference-plus-Noise Ratio (SINR) is defined as the ratio of

the desired signal power divided by the undesired signal power

$$SINR = \frac{P_{ss}}{P_{uu}} = a^2 \frac{\left|\bar{w}^H \cdot \bar{x}_s\right|^2}{\bar{w}^H \cdot \bar{R}_{uu} \cdot \bar{w}} \tag{6.7}$$

Maximizing the SINR is a criterion which can be applied to enhancing the received signal while minimizing the interfering signals [25]. The Applebaum optimization criterion [26], consists in maximizing SINR. But a direct maximization of Equation 6.7 is not possible since neither a nor $\bar{R}_{uu}$ can be directly measured. However, as shown in [27], the equation can be recast as the maximization of the fitness function $f(\bar{\omega})$

$$f(\bar{\omega}) = \frac{\left|\bar{\omega}^H \cdot \bar{x}_s\right|^2}{\bar{\omega}^H \cdot \bar{R}_{xx} \cdot \bar{\omega}} \tag{6.8}$$

Equation 6.8 can be used as the fitness function for population-based optimization algorithms like the Particle Swarm Optimization and Genetic Algorithms.

6.3 Results

In this section, the capabilities of the PSO algorithm to obtain an optimal set of weights are assessed. The results of several numerical experiments are presented and compared with those of the GA algorithm.

Three different experiments are conducted: The first experiment involves the optimization of a 10x10-element rectangular array. The aim is to use the SINR in the fitness function of the Particle Swarm Optimization and the Genetic Algorithm to find a set of array weights. These weights should configure the antenna array in such a way that it can receive desired signals and block undesired or jammer signals. The results are presented as graphics showing the resulting radiation patterns. Another goal of this experiment is to calculate the number of evaluations performed by each of the two algorithms. This figure is useful to predict the overall performance of a communication system, as often the computational effort happens when evaluating the fitness function. Thus, reducing the number of evaluations, while obtaining an acceptable result is important for mobile communication devices.

The second experiment is similar to the first one. The PSO and GA algorithms find the antenna set of weights but this time using a fixed number of iterations. The goal of the experiment is to evaluate the minimum gain level towards the prescribed nulls obtained by each algorithm. This experiment helps to assess the capability of the PSO and GA algorithms to find the best possible solution given a limited number of fitness function evaluations.

The third and last experiment consists in solving the same problem as before but for different antenna array configurations (20-element array, 2x8-element array, 4x6-element array, 5x5-element array and 10x10-element array). In this experiment, the algorithms Least Means Squares (LMS) and Recursive Least Squares (RLS) [92] together with the PSO and GA algorithms, solve the problem for each antenna geometry.

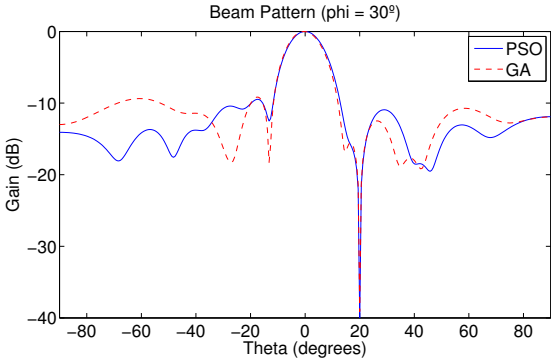
6.3.1 Experiment 1: Number of evaluations

The first experiment deals with the optimization of a 10×10-element half-wavelength-spaced planar rectangular array. The goal is to find a set of array weights that will configure each element to obtain an optimized beam pattern for the following conditions: A desired transmitter called $user1$ at the direction (θ=0°,ϕ=0°) and 3 undesired signals $jammer1$, $jammer2$ and $jammer3$ coming from directions (θ=20°,ϕ=30°), (θ=60°,ϕ=45°) and (θ=-40°,ϕ=-60°) respectively.

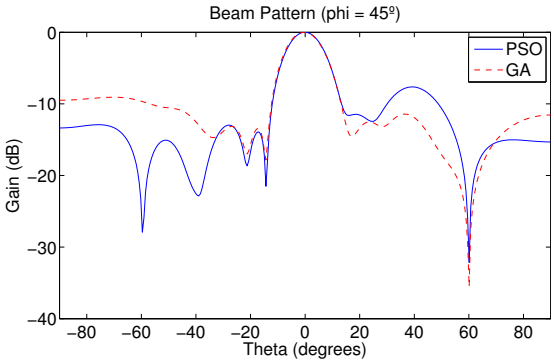
To solve this problem, the PSO and GA algorithms are initialized with the following parameters: Dimension $= M \times N = 10 \times 10 = 100$. Number of individuals = 24. The initial positions of the individuals are randomly generated inside the search space with minimum and maximum boundaries of $[-1, 1]$. The stopping criteria for both algorithms consists of a tolerance value of 1×10^{-4}. In other words, the process stops when the fitness function reaches a value greater than or equal to $1 - (1 \times 10^{-4})$.

Table 6.1 shows the parameters set for the PSO and GA algorithms for this experiment. Note that both algorithms use the same tolerance (1×10^{-4}) in order for them to exert the same effort to find a solution.

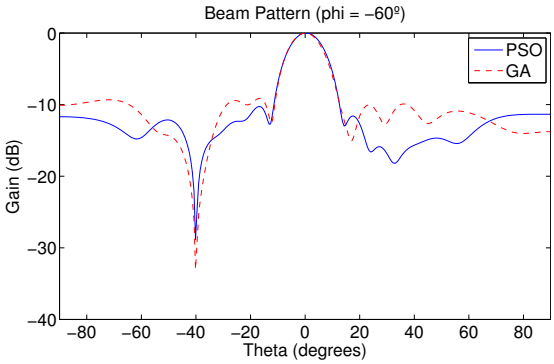
The PSO algorithm was programmed in Matlab and is based on the Standard PSO 2007 proposed by Maurice Clerc in [93]. The PSO parameters are set as



(a) Radiation pattern for $jammer1$ (θ=20°,ϕ=30°).



(b) Radiation pattern for $jammer2$ (θ=60°,ϕ=45°).



(c) Radiation pattern for $jammer3$ (θ=-40°,ϕ=-60°).

Figure 6.2: PSO and GA radiation patterns for 3 jammers.

Algorithm	PSO	GA
Dimension	$M \times N = 10 \times 10 = 100$	
Number of individuals	24	
Initial position	$[-1, 1]$	
Minimum boundary	-1	
Maximum boundary	1	
Tolerance	1×10^{-4}	
Inertia weight (w)	1.0	-
Correction factor ($c1$, $c2$)	2.0	-
Individual's encoding	-	Real
Selection	-	Stochastic uniform
Mutation	-	Constraint dependent default
Crossover	-	Scattered

Table 6.1: Parameters used by the PSO and GA algorithms when optimizing a 10×10-element planar rectangular array

following: The weight w in Equation 3.1, also called inertia, is set to 1.0. The constants c_1 and c_2, or correction factor, are set to 2.0. Previous work has shown that this value is a good choice for both parameters [86]. For the case of the GA algorithm, the "Matlab Genetic Algorithm and Direct Search Toolbox" is used. All parameters of the *ga* Matlab function are left to their default values except for the ones previously mentioned. Note that by using the same tolerance as the PSO (1×10^{-4}) both algorithms exert the same effort to find a solution.

After running the simulations, each algorithm obtained a vector of array weights. Figure 6.2 shows the resulting radiation patterns in dB (decibels). It can be seen that both algorithms are able to generate suitable array weights to produce the desired radiation pattern. The main lobe, that corresponds to $user1$ is directed towards (θ=0°,ϕ=0°), whereas 3 nulls appear for the undesired signals $jammer1$ = (θ=20°,ϕ=30°), $jammer2$ = (θ=60°,ϕ=45°) and $jammer3$ = (θ=-40°,ϕ=-60°). In order to obtain statistical significance, the algorithms are run 100 times. The mean was calculated and the results are shown in Table 6.2.

Mean for 100 runs	PSO	GA
Number of evaluations	327.36	960
Number of iterations	14.64	39
Area above the pattern	32.9896	27.8802

Table 6.2: PSO and GA statistics after 100 runs.

It can be observed that the PSO executes a lower number of fitness function

evaluations on average. The number of evaluations helps to predict the overall performance of the system, given that most of the computational effort done by the algorithm consists of the evaluation of the fitness function. By calculating the area above each radiation pattern, it can be noted that the sidelobe Levels (SLL) which are the lobes other than the main lobe, are lower for the PSO compared to those for the GA. This is often desirable as it helps to avoid other interfering signals at different directions other than the main lobe.

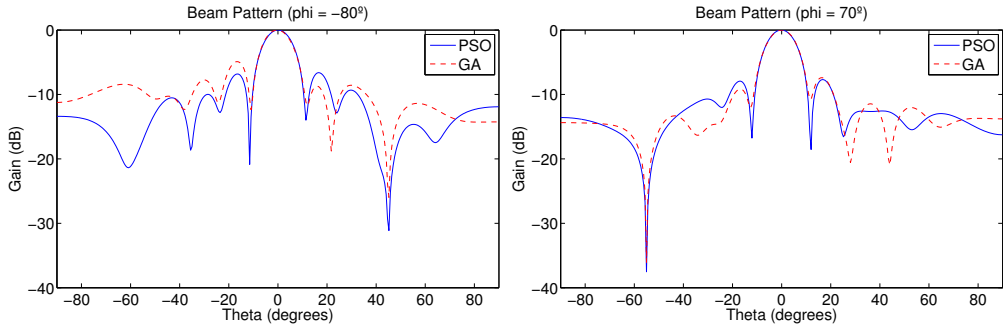
6.3.2 Experiment 2: Gain level of prescribed nulls

A second experiment was performed, this time with 4 interferers: $jammer1$, $jammer2$, $jammer3$ and $jammer4$ coming from directions (θ=45°,ϕ=-80°), (θ=-55°,ϕ=70°), (θ=-20°,ϕ=50°) and (θ=80°,ϕ=40°) respectively. The same parameters of the previous experiment are used except for the stopping criteria. This time the number of iterations is fixed to 100 for both GA and PSO. The purpose of this change is to evaluate the minimum gain level reached in the direction of prescribed nulls. Figure 6.3 shows the radiation patterns obtained.

The mean minimum power at all the nulls for the PSO algorithm is -35.7 dB whereas the power for the GA is -23.4 dB. These figures suggest that the PSO algorithm tends to perform better than the GA algorithm in terms of radiation power directed towards undesired signals.

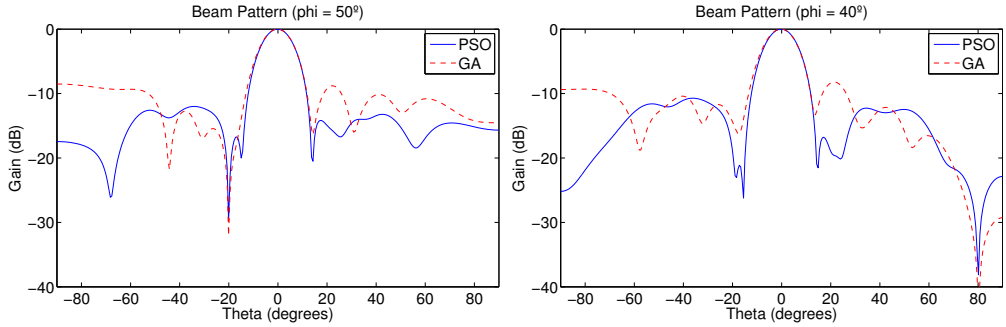
6.3.3 Experiment 3: Different array configurations

In the third and last experiment, the Particle Swarm Optimization, the Genetic Algorithm, the Least Means Squares and the Recursive Least Means are used to solve a similar problem to the previous experiments. Three interferers: $jammer1$, $jammer2$, and $jammer3$ coming from directions (θ=-60°), (θ=-30°) and (θ=45°) are to be avoided. Figure 6.7 shows the array pattern of a 20-element linear antenna array. It can be seen that the PSO algorithm obtains the lowest null levels compared with the GA, LMS and RLS methods. Figures 6.8, 6.9, 6.10 and 6.11 show similar results for a 2x8-element array, 4x6-element array, 5x5-element array and 10x10-element array respectively. These results show that both the



(a) Radiation pattern for *jammer*1 (θ=45°,ϕ=-80°).

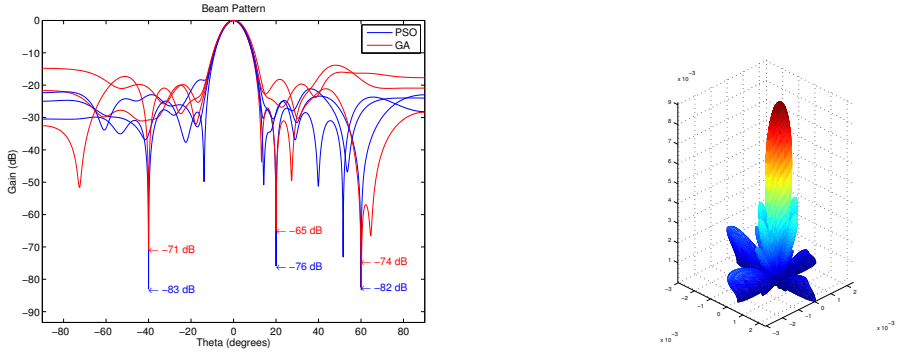
(b) Radiation pattern for *jammer*2 (θ=-55°,ϕ=70°).



(c) Radiation pattern for *jammer*3 (θ=-20°,ϕ=50°).

(d) Radiation pattern for *jammer*4 (θ=80°,ϕ=40°).

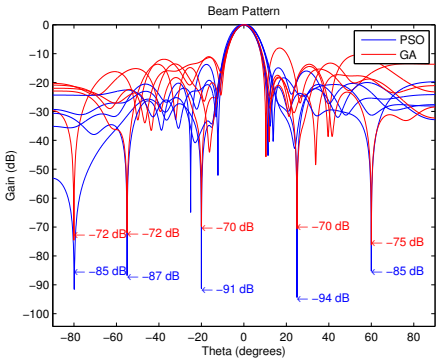
Figure 6.3: PSO and GA radiation patterns for 4 jammers.



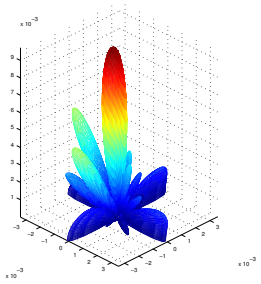
(a) PSO and GA 2D array pattern.

(b) PSO 3D array pattern.

Figure 6.4: 3 interferers.



(a) PSO and GA 2D array pattern.



(b) PSO 3D array pattern.

Figure 6.5: 5 interferers.

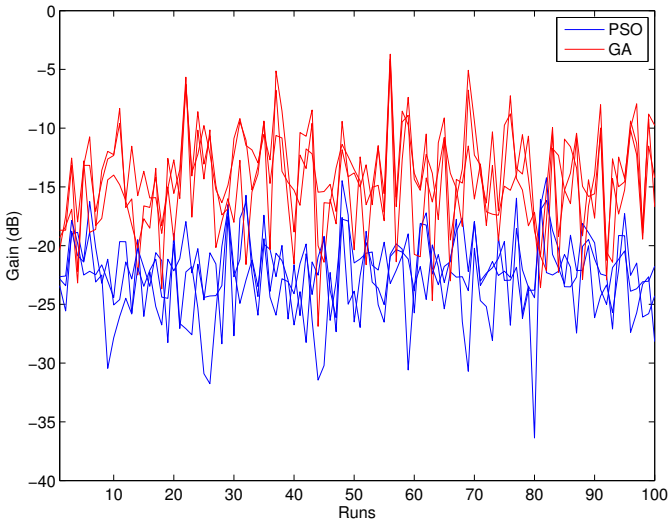


Figure 6.6: Null levels for each run.

PSO and GA algorithms are capable of obtain the desired results independently from the geometry configuration of a given antenna array.

6.4 Summary

In this chapter, the Signal-to-Interference-plus-Noise Ratio was used in the fitness function for population-based optimization algorithms. The Particle Swarm Optimization method was used to generate a set of array weights to configure a planar rectangular array. These weights were optimized in order to maximize the power towards a desired direction whilst minimizing it in the direction of interferers. A standard Genetic Algorithm was also studied and the results show that the PSO performs better in terms of the total number of fitness function evaluations. This suggests an advantage in performance as the fitness function tends to have a high computational cost. It was also observed that the PSO obtains on average lower sidelobe levels which are desirable to avoid interference. Furthermore, the gain levels in the direction of nulls were computed and it was found that the PSO produces lower values than those of the GA.

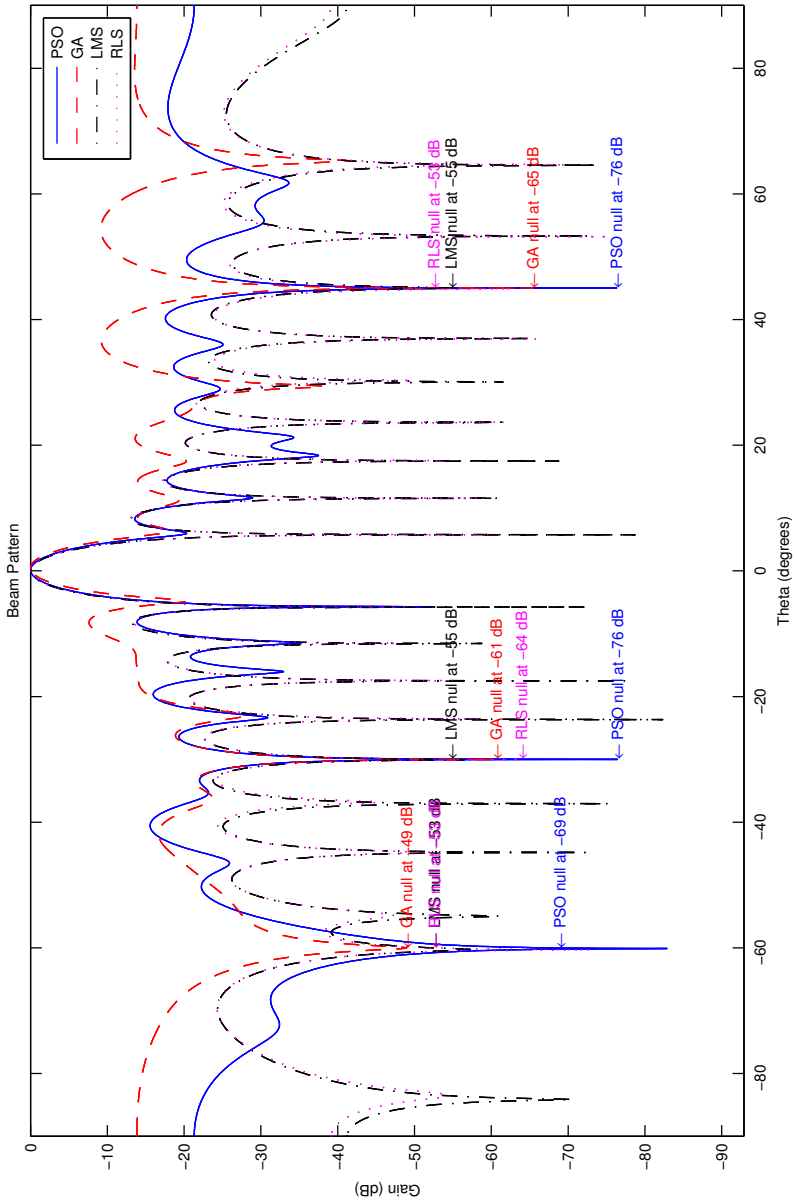


Figure 6.7: Linear 20-element array pattern showing nulls at -60°, -30° and 45° for algorithms PSO, GA, LMS and RLS.

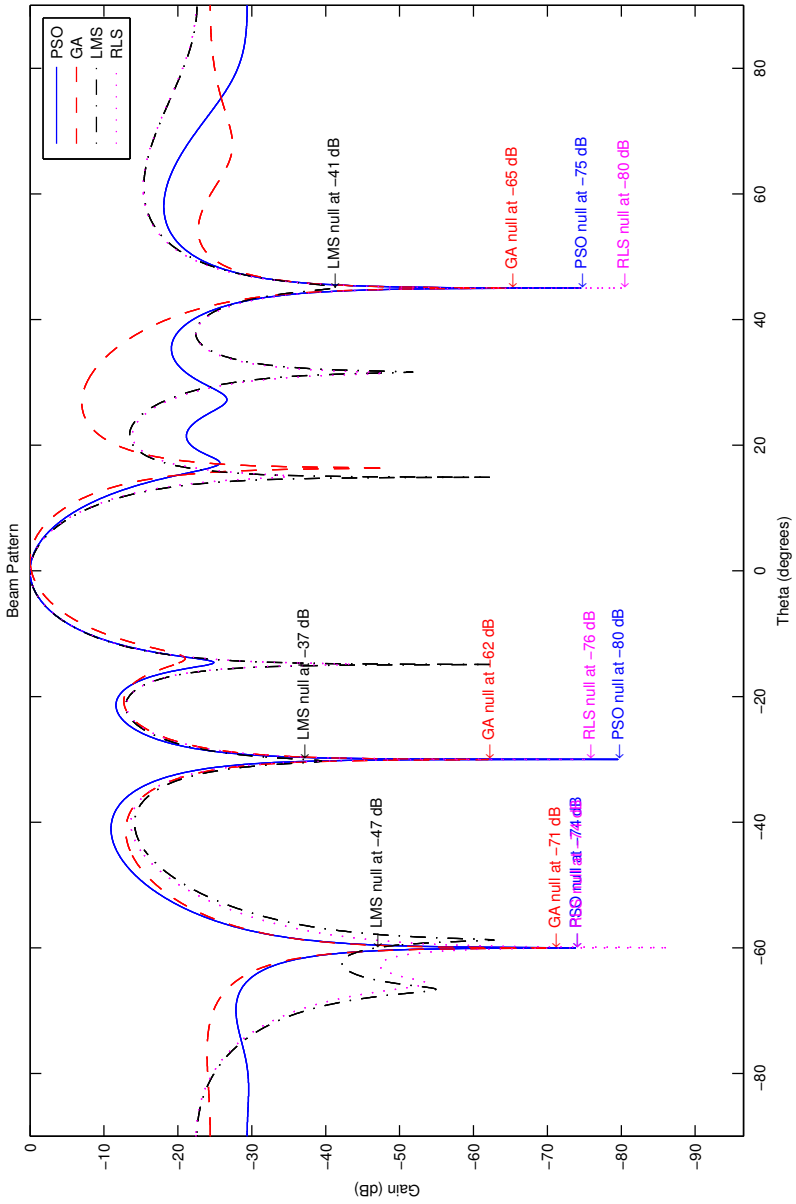


Figure 6.8: Rectangular 2x8-element array pattern showing nulls at -60°, -30° and 45° for algorithms PSO, GA, LMS and RLS.

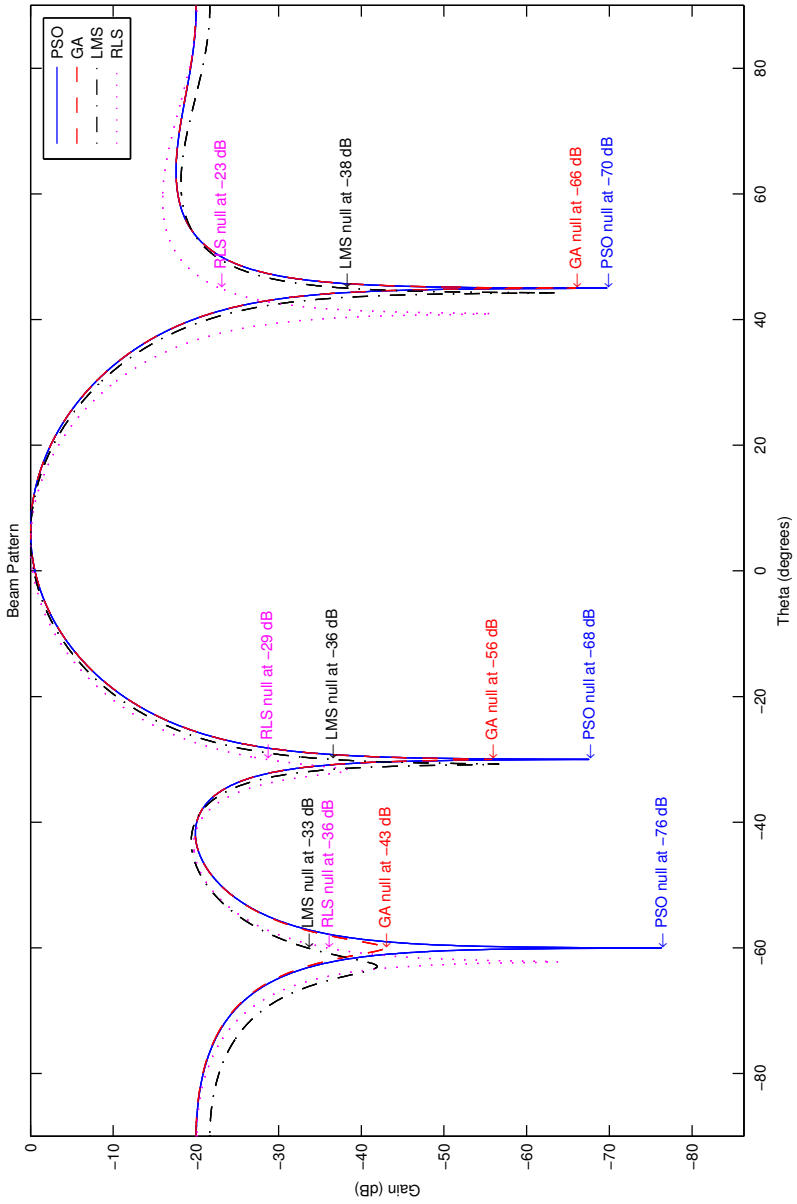


Figure 6.9: Rectangular 4x6-element array pattern showing nulls at -60°, -30° and 45° for algorithms PSO, GA, LMS and RLS.

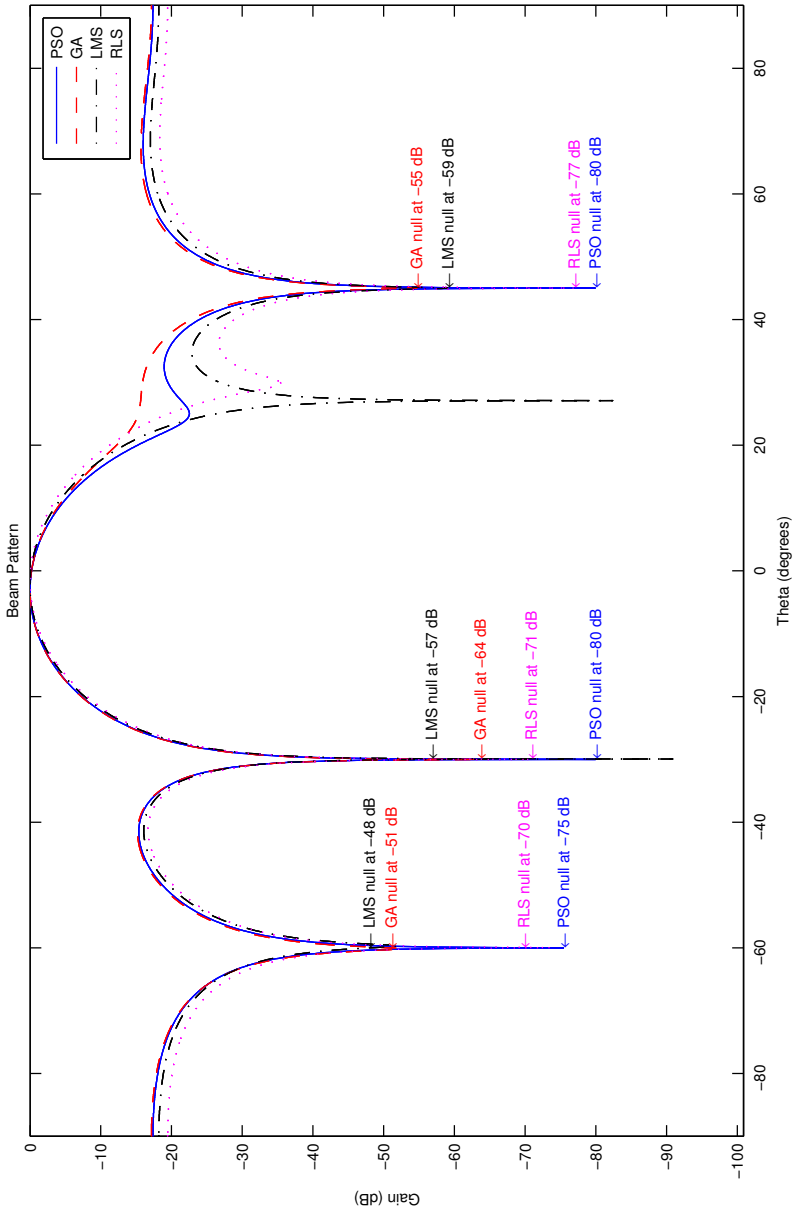


Figure 6.10: Rectangular 5x5-element array pattern showing nulls at -60°, -30° and 45° for algorithms PSO, GA, LMS and RLS.

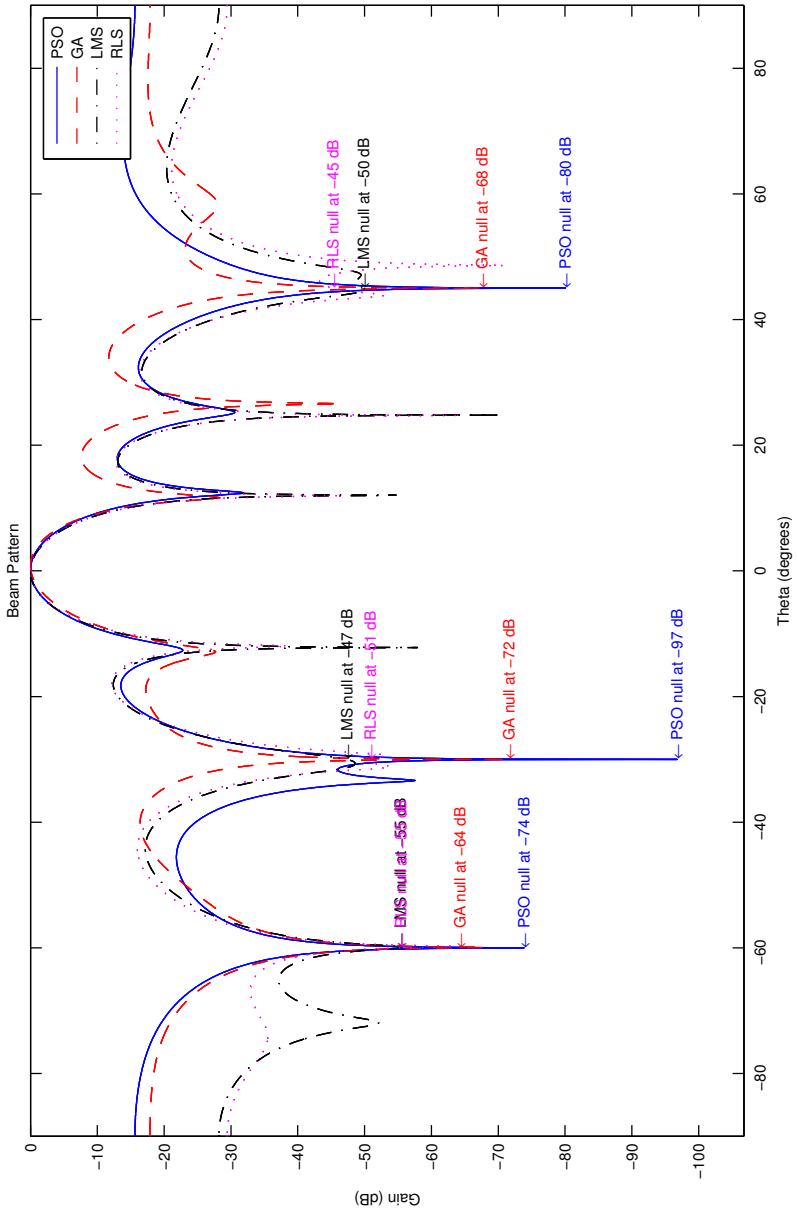


Figure 6.11: Rectangular 10x10-element array pattern showing nulls at -60°, -30° and 45° for algorithms PSO, GA, LMS and RLS.

Chapter 7

Meta-Optimization Techniques

During the development of this thesis it was observed that the performance of optimization algorithms depends heavily on their initial parameters. To enhance the effectiveness of an algorithm, these parameters should be carefully selected according to the problem to be solved. For example, the GA has crossover and mutation rates which will affect the overall ability of the algorithm to converge to the desired solution. By modifying these parameters, a good balance between exploration and exploitation can be achieved. In this chapter, the parameters of algorithms such as Differential Evolution, Simulated Annealing, Hill Climb and Particle Swarm Optimization will be selected using a technique called meta-optimization. This process consists of using another optimization algorithm to find good behavioural parameters. A group of algorithms, namely Pattern Search, Local Unimodal Sampling as well as DE and PSO are selected to act as a second layer of optimization over the mentioned techniques. Meta-landscapes, which are graphical representations of the meta-optimization problem are shown as well as statistics obtained for each meta-optimization experiment. A similar antenna problem to that in the previous chapter is solved using the obtained meta-optimized parameters. Moreover, antenna synthesis problems proposed in the literature will also be tackled using meta-optimization techniques. Results show that these meta-optimized parameters enable the algorithms to obtain better results than those achieved by standard parameters as well as the parameters obtained by hand-tuning.

7.1 Introduction

Traditionally, the behavioural parameters have been chosen according to numerous experiments done by researchers. One example is presented in [94], where the impact of inertia weight and maximum velocity on the performance of the PSO algorithm is analysed. A number of experiments were performed with different values for these parameters and it was concluded that when the maximum velocity is small, an inertia weight of approximately 1 is a good choice. Another example of parameter analysis is given in [95], where a constriction factor is proposed to limit the maximum velocity while using the inertia weight according to a given equation. In relation to the DE algorithm, Storn et al. [36] describe several variants of the algorithm and provide some general hints on their usage.

Parameters can also be selected according to mathematical analysis as shown in [12] in which the PSO algorithm is analyzed and graphical parameter selection guidelines are provided. This study showed different results of the speed of convergence and the treadeoff with the robustness of the solutions. Clerc and Kennedy [93] also show an analytical view of the particle's trajectory which leads to a generalized model of the algorithm and its convergence tendencies.

The selection of parameters can be divided into two cases: parameter tuning and parameter control [13]. In parameter control the parameter values change during the optimization run. An initial parameter value is needed and it has to suit the control strategies that can be deterministic, adaptive, or self-adaptive [14]. On the other hand, in parameter tuning the values do not change during the run but there is still a large number of combinations depending on the number of parameters (variables). The following section explains the meta-optimization strategy which is a kind of parameter tuning.

7.2 Meta-optimization

Meta-optimization consists of using an optimization method to tune the parameters of another optimization method. Meta-optimization is also called Meta-evolution or Automated Parameter Calibration. This concept was used as early as 1978 by Mercer and Sampson [96], but due to the large computational costs,

their research was very limited. In 1986 Grefenstette [97] applied a GA as a second level optimizer to identify efficient GAs for a set of numerical optimization problems. As explained in the previous section, optimization techniques such as PSO and GAs have a set of parameters that control their behaviour when optimizing a given problem. These parameters affect greatly the output of the optimization method and must be chosen carefully. It is worth mentioning that a given set of parameters could work well when optimizing a specific problem but perform differently when optimizing another. Thus, finding the best parameters can be an arduous task and will depend on human perception of how they work.

Meta-optimization allows an objective way to find the most suitable set of parameters for a given optimization method and problem to be solved. The way meta-optimization works is by using an optimization algorithm that has the parameters as output. During the meta-optimization process, every new set of parameters is used by the optimization algorithm and its output evaluated. Thus, the outer layer is in charge of finding a better set of parameters until a stop condition is met. Figure 7.1 shows this concept.

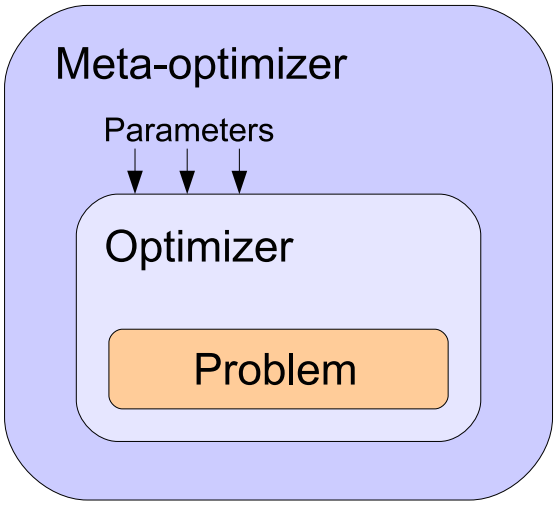


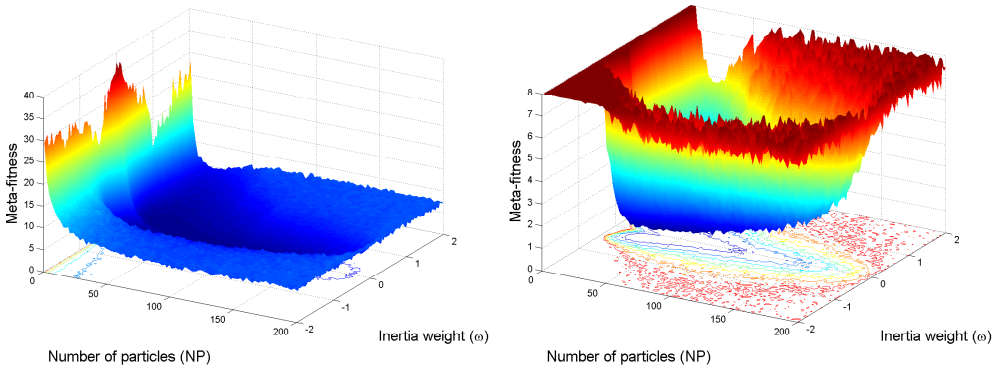
Figure 7.1: Meta-optimization. The parameters of the optimization algorithm are obtained by another optimization algorithm as a second layer.

7.3 Meta-landscapes

In optimization problems, researchers often have to deal with the complexity of having an exponential increase of dimensions in a problem space. This is called the *curse of dimensionality*, a term coined by Richard E. Bellman [98]. In meta-optimization, the number of parameters to be optimized is multiplied by the meta-optimizer's own parameters. For each new parameter, the number of candidate solutions grows exponentially. For this reason, it is desirable that the meta-optimizer algorithm is as simple as possible. This means that the problem to be solved by the meta-optimizer should not be so complex as to have the algorithm fail to converge. One way to observe the kind of problem the meta-optimizer has to deal with, is to generate a 3-dimensional plot of the meta-fitness landscape. Most of the methods have more than two parameters so it is necessary to fix some of them to be able to produce a viewable plot.

Consider the PSO algorithm with c_1 and c_2 fixed to 1.49445. The variable parameters being the number of particles NP and the inertia weight ω. The boundaries for these parameters are [1, 200] for NP and [-2, 2] for ω. Figure 7.2(a) shows the meta-fitness landscape of the PSO when optimizing a 16-element linear antenna array and using the SINR (Signal to Interference plus Noise Ratio) as the fitness function. The algorithm is executed 50 times and runs for 200 iterations. These results suggest that the problem of meta-optimizing the PSO algorithm to solve the antenna array is simple. The meta-landscape surface is fairly regular and without obvious local minima. The graph shows that the meta-fitness values are worst when only a few number of particles are used. Figure 7.2(b) shows a closer look at the meta-landscape by capping the meta-fitness values at 8. It can be observed that a larger number of particles (above 50) is not only unnecessary but also worsens the results. In addition there is a symmetry in the values of ω, both of which are better when they are close to 0, either with positive or negative values. Something similar can be observed in Figure 7.3(a) where the meta-landscape for the DE algorithm was obtained with the same problem. In this case, the variables are the number of particles NP and the crossover probability CR. The third parameter F is fixed to 0.6. It can also be noticed that the meta-landscape is fairly simple with a single minima. Figure 7.3(b) shows a close up of this region.

In the same fashion, the meta-landscapes for the SA and HC algorithms were obtained and are shown in Figures 7.4 and 7.5 respectively. For the SA algorithm the sampling range r and the start value α were used as variables while the end value β was set to 0.01 and the temperature T to 40000. The HC algorithm has only two parameters: the sampling range factor r and the probability weight D. Both figures show meta-lanscapes quite similar to those of the DE and PSO algorithms with a valley of good performing parameter combinations. It should be recalled that for these experiments, some of the parameters were fixed, so it is possible that an optimization using all the parameters will look different. The full results will be shown in the following sections.

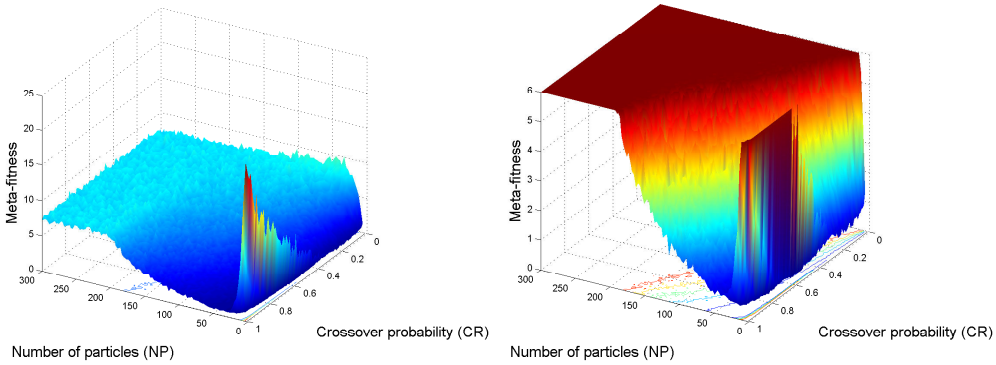


(a) Meta-landscape for 2 dimensions. (b) Capped at 8 for clarity.

Figure 7.2: Meta-landscape for PSO obtained by varying two dimensions: The number of particles and the weight. The third and fourth parameters (c_1 and c_2) are fixed to 1.49445. For 50 runs and 200 iterations. 16-element linear antenna array.

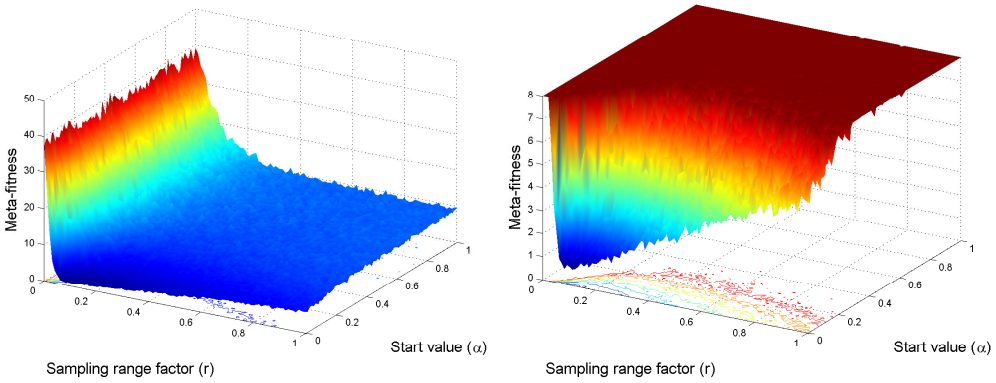
7.4 Parameter Tuning of Optimization Methods for Antenna Arrays

As explained in previous sections, a meta-optimization technique can be applied to optimization algorithms solving antenna problems. In Chapter 6 the PSO and GA algorithms were used to maximize the SINR to enhance the received signal while minimizing the interferers of an adaptive antenna array. A set of weights were obtained as a result of the PSO and GA techniques. In this section



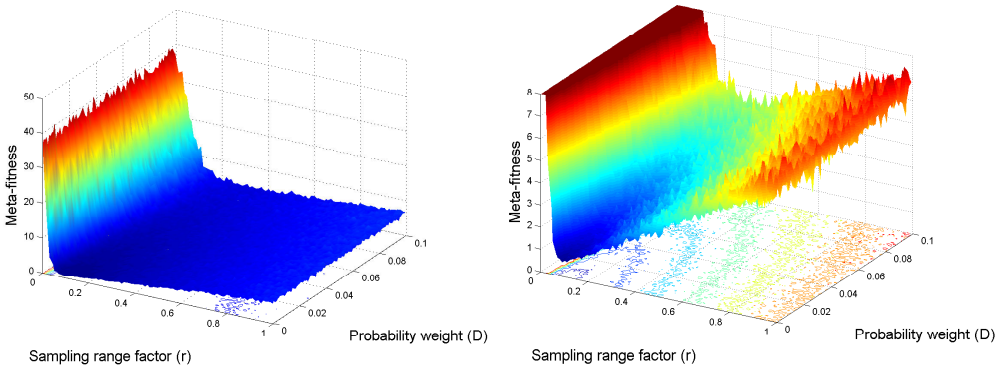
(a) Meta-landscape for 2 dimensions. (b) Capped at 6 for clarity.

Figure 7.3: Meta-landscape for DE obtained by varying two dimensions: The number of particles and crossover probability. The third parameter F is fixed to 0.6. For 50 runs and 200 iterations. 16-element linear antenna array.



(a) Meta-landscape for 2 dimensions. (b) Capped at 8 for clarity.

Figure 7.4: Meta-landscape for SA obtained by varying two dimensions: The sampling range factor and α. The third and fourth parameters β and T are fixed to 0.01 and 40000 respectively. For 50 runs and 200 iterations. 16-element linear antenna array.



(a) Meta-landscape for 2 dimensions. (b) Capped at 8 for clarity.

Figure 7.5: Meta-landscape for HC obtained by varying two dimensions: The sampling range factor r and probability weight D. For 50 runs and 200 iterations. 16-element linear antenna array.

the Differential Evolution, Simulated Annealing and Hill Climb algorithms will be used along with the PSO. Howerver before optimizing an antenna problem, their set of parameters will be meta-optimized using simple algorithms like Pattern Search and Local Unimodal Sampling. The DE and PSO will also be used as meta-optimizers. The following subsections describe the meta-optimization process.

7.4.1 Meta-optimization Results

The problem to be solved consists of a 16-element uniform linear antenna array with a distance between elements of 0.5λ. A desired signal arrives from the $0°$ direction while three interferer signals come from $30°$, $60°$and $-45°$. In order to obtain the weights needed, Equation 6.8 will be used by the optimization process as the fitness function.

The meta-optimization algorithms were programmed in C language using a modified version of *SwarmOps* which is a source-code library for doing numerical optimization written by Pedersen [99]. The *SwarmOps* source-code is published under the GNU Lesser General Public License. Table 7.1 shows the results of the meta-optimization process which was run to obtain the best parameters for the DE algorithm. The first 4 columns show the four algorithms used as meta-

optimizers as well as the parameters used. PS needs no initial parameters, LUS uses $\gamma = 3$ while DE and PSO have a set of parameters (NP=50, CR=0.9, F=0.6) and (S=50, ω=0.729, c_1=c_2=1.4944) respectively. It is worth noting that these parameters are taken from [35, 100] for DE and [86, 101] for PSO and will be considered the "standard" ones in this thesis. Another factor in the meta-optimization process is the number of meta-runs which in this case is 6. The number of meta-iterations is defined according to the number of parameters to be optimized multiplied by 20. This will allow a fair number of meta-iterations depending on the number of variables. The fifth column in Table 7.1 is the number of optimization iterations performed in the meta-optimization phase. In this case, for each process three different experiments were run: using 200, 500 and 1000 iterations. This is to observe the variation in the quality of the results according to the number of iterations and it will be discussed later on. The next column is the meta-fitness obtained in every experiment and shows the value obtained at the end of the 60 meta-iterations. The last three columns present the output which is the set of parameters suitable for the DE algorithm to best solve the antenna problem. The first is the number of particles NP, the second is the crossover probability CR and the third the differential weight F as described in Chapter 3. These set of parameters will be used in the second phase of the optimization problem which is to run the DE algorithm to optimize the linear array described above. Another important factor in the meta-optimization process is the number of runs or repetitions performed by the algorithm being meta-optimized (in this case the DE). This is in order to obtain statistical significance and is set to 100 runs. In other words, several runs are performed to minimize the chance that a better or worse result is obtained due only to the random nature of the algorithms. For these reason and, as explained in previous sections, the meta-optimization process is as complex as the multiplication of its stages, thus, the time taken by the experiment to finish can be considered lengthy. Table 7.5 shows the time taken in hours by each of the four meta-optimizers when running the optimized algorithm for 200, 500 and 1000 iterations. The description of the computer system used to run all the simulations on this thesis is the following: 64-bit Intel 4-Core i7 720QM @ 1.60GHz with 4.0Gb in RAM. Note: It should

be mentioned that the nature of the processor, which has 4 cores, provided a substantial reduction in execution time compared with other architectures, for example with 1 core only. The meta-optimization of the algorithms SA, HC and PSO was performed in the same fashion. The only difference is the number of meta-iterations, given that each algorithm has a different number of parameters. The results for these algorithms are shown in Tables 7.2, 7.3 and 7.4 respectively.

7.4.2 Optimization Results

Once the meta-optimized parameters for each algorithm are obtained, the next step is to run the algorithms using those parameters. In order to assess the results of the meta-optimization process, the results are shown together with the results of simulations performed with standard as well as hand-tuned parameters. The parameters shown in Tables 7.1, 7.2, 7.3 and 7.4 are then used to configure the DE, SA, HC and PSO algorithms in order to solve the same problem as the previous section. Note that not only the results from using standard parameters are shown, but the ones resulting from hand-tuning or trial an error of adjusting the parameters to solve this particular antenna problem. The parameters (Refer to Chapter 3 for a detailed explanation for each parameter) used in these experiments are shown in Tables 7.6, 7.7, 7.8 and 7.9. A close look at Table 7.6 shows that the number of particles NP suggested by the meta-optimization process is always less than the standard and the hand-tuned data. This can be regarded as an advantage if an implementation in hardware is in mind. For example, in a cellular architecture where each processing element implements a particle or individual, this reduction will lead to a decreased power consumption. The same situation can be observed in the case of the PSO algorithm in Table 7.9.

7.4.3 PSO Particle Velocity and Position

One way to assess the performance of the algorithms using the meta-optimized parameters is to observe their behaviour. In the case of the PSO, the particles have a velocity and a position. Each particle is given a random velocity and a random position at the beginning of the algorithm. In the next iteration, each

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization Iterations	Meta-fitness	Best found parameters		
						NP	CR	F
PS	N/A	6	60	200	0.307881	36	0.937561	0.241292
				500	0.200027	29	0.812863	0.434416
				1000	0.166988	78	0.992188	0.500000
LUS	$\gamma = 3$			200	0.321626	32	0.891637	0.270391
				500	0.217811	38	0.707868	0.245491
				1000	0.157331	67	0.987485	0.297393
DE	NP = 50 CR = 0.9 F = 0.6			200	0.351874	36	0.888624	0.201017
				500	0.223375	28	0.762065	0.282750
				1000	0.175920	20	0.854080	0.574468
PSO	S = 50 $\omega = 0.729$ $c_1, c_2 = 1.4944$			200	0.501596	31	0.854729	0.140400
				500	0.214974	12	0.430857	0.377390
				1000	0.167921	98	0.972746	0.163822

Table 7.1: Meta-optimization of the Differential Evolution parameters using meta-methods PS, LUS, DE, PSO and number of iterations 200, 500 and 1000. The problem being solved is a 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization Iterations	Meta-fitness	Best found parameters			
						r	α	β	T
PS	N/A	6	80	200	0.432773	0.0521403	$0.010e^{-3}$	0.000376	6343.75
				500	0.263451	0.0152584	$0.010e^{-3}$	0.593754	$100.00e^{3}$
				1000	0.205891	0.0160618	$0.269e^{-3}$	1.000000	55289.60
LUS	$\gamma = 3$			200	0.499939	0.0537584	$0.440e^{-3}$	0.322461	82547.10
				500	0.541506	0.1076860	$1.026e^{-3}$	0.927442	96649.30
				1000	0.323078	0.0697977	$2.196e^{-3}$	0.589459	10034.30
DE	NP = 50 CR = 0.9 F = 0.6			200	0.468874	0.0520060	$0.010e^{-3}$	0.019163	08158.36
				500	0.303287	0.0300787	$0.010e^{-3}$	0.764745	23695.00
				1000	0.252692	0.0176584	$0.010e^{-3}$	1.000000	90957.70
PSO	S = 50 $\omega = 0.729$ c_1, $c_2 = 1.4944$			200	0.480620	0.0527300	$0.010e^{-3}$	0.670021	51376.20
				500	0.292168	0.0180429	$0.010e^{-3}$	0.123615	88774.50
				1000	0.249360	0.0076126	$0.010e^{-3}$	0.844229	$100.00e^{3}$

Table 7.2: Meta-optimization of the Simulated Annealing parameters using meta-methods PS, LUS, DE, PSO and number of iterations 200, 500 and 1000. The problem being solved is a 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization Iterations	Meta-fitness	Best found parameters	
						r	D
PS	N/A	6	40	200	0.48097	0.039062	$1.00e^3$
				500	0.28095	0.023437	$1.00e^3$
				1000	0.24452	0.035156	$1.00e^3$
LUS	$\gamma = 3$			200	13.9533	0.991528	2573.24
				500	9.94216	0.638403	9684.70
				1000	8.14475	0.720823	5312.66
DE	NP = 50 CR = 0.9 F = 0.6			200	13.8622	0.887655	0778.25
				500	10.1865	0.734335	3010.22
				1000	7.93402	0.987598	2841.82
PSO	S = 50 $\omega = 0.729$ c_1, $c_2 = 1.4944$			200	13.8162	0.728137	2973.91
				500	9.96688	0.744350	6768.48
				1000	7.93787	0.895648	2671.36

Table 7.3: Meta-optimization of the Hill Climb parameters using meta-methods PS, LUS, DE, PSO and number of iterations 200, 500 and 1000. The problem being solved is a 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

Meta-method	Meta-parameters	Meta-runs	Meta-iterations	Optimization Iterations	Meta-fitness	Best found parameters			
						S	ω	c_1	c_2
PS	N/A	6	80	200	0.0731948	39	0.006882	-0.062488	1.7290
				500	0.0655238	40	0.060813	0.306313	2.1190
				1000	0.0644712	50	0.139359	0.492094	2.5713
LUS	$\gamma = 3$			200	0.0860273	42	-0.014691	0.221414	1.2760
				500	0.0653379	39	-0.155383	0.117276	1.7069
				1000	0.0640246	43	0.457861	1.000290	1.0520
DE	NP = 50 CR = 0.9 F = 0.6			200	0.0787030	40	-0.008263	0.103120	1.4287
				500	0.0667847	21	0.418945	1.110040	1.7152
				1000	0.0642280	44	0.427277	1.242550	1.0825
PSO	S = 50 $\omega = 0.729$ c_1, $c_2 = 1.4944$			200	0.1040050	43	-0.010369	0.010173	3.1111
				500	0.0686567	35	0.008913	0.131063	2.8173
				1000	0.0641744	41	-0.312289	0.115648	1.8023

Table 7.4: Meta-optimization of the Particle Swarm Optimization parameters using meta-methods PS, LUS, DE, PSO and number of iterations 200, 500 and 1000. The problem being solved is a 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

Meta-method	Time usage (hrs)		
	200 iterations	500 iterations	1000 iterations
PS	0:46	2:31	4:27
LUS	0:34	1:57	3:37
DE	0:49	2:35	4:37
PSO	0:48	2:34	4:38

Table 7.5: Time taken by each meta-method to optimize the four algorithms DE, SA, HC and PSO running on a 64-bit Intel 4-Core i7 720QM @ 1.60GHz system. Each algorithm was run 100 times.

Meta-method	NP	CR	F
Standard [35, 100]	300	0.9	0.5
Hand-tuned	50	0.01	0.05
PS	36	0.937561	0.241292
LUS	32	0.891637	0.270391
DE	36	0.888624	0.201017
PSO	31	0.854729	0.140400

Table 7.6: Standard, hand-tuned and meta-optimized Differential Evolution parameters used to solve the 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

Meta-method	r	α	β	T
Standard	0.01	0.3	0.01	40000
Hand-tuned	0.05	0.03	0.5	50000
PS	0.05214	0.00001	0.000376	6343.75
LUS	0.05375	0.00044	0.322461	82547.10
DE	0.05200	0.00001	0.019163	8158.36
PSO	0.05273	0.00001	0.670021	51376.20

Table 7.7: Standard, hand-tuned and meta-optimized Simulated Annealing parameters used to solve the 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

particle calculates a new velocity and a new position according to the PSO equations 3.1 and the fitness function. The expected behaviour is that the particles move faster in the early stages of the process and reduce their velocity as they get closer and closer to the solution. This helps the algorithm to converge faster

Meta-method	r	D
Standard	0.01	10
Hand-tuned	0.5	5000
PS	0.039063	0.001
LUS	0.991528	2573.24
DE	0.887655	778.25
PSO	0.728137	2973.91

Table 7.8: Standard, hand-tuned and meta-optimized Hill Climb parameters used to solve the 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

Meta-method	S	ω	c_1	c_2
Standard [86, 101]	50	0.729	1.49445	1.49445
Hand-tuned	50	0.09	1	1
PS	39	0.006883	-0.062488	1.72909
LUS	42	-0.014692	0.221414	1.27601
DE	40	-0.008264	0.10312	1.42876
PSO	43	-0.010369	0.010174	3.11112

Table 7.9: Standard, hand-tuned and meta-optimized Particle Swarm Optimization parameters used to solve the 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45° .

as the probability of a particle being initialized close to the solution is low but increases with the iterations as it improves the solution.

In Figure 7.6, the velocity of one of the particles over the number of iterations is shown. Each graph represents the velocity when using standard, hand-tuned, PS and LUS meta-optimized parameters. It can be seen that the velocity of the particle with the standard parameters varies in the range of [-1, 1] along the iterations and keeps changing even when reaching 1000 iterations. This means that the PSO is unable to converge at this moment when using the standard parameters. The second graph shows the velocity with hand-tuned parameters. This time, the velocity is reduced and reaches 0 before 400 iterations. Then, when using the LUS and PS meta-optimized parameters the velocity settles significantly faster, even before 200 iterations. These results show that the meta-optimization has enabled the PSO algorithm achieve better performance in terms of convergence

time and accuracy which can be very helpful in real-time systems.

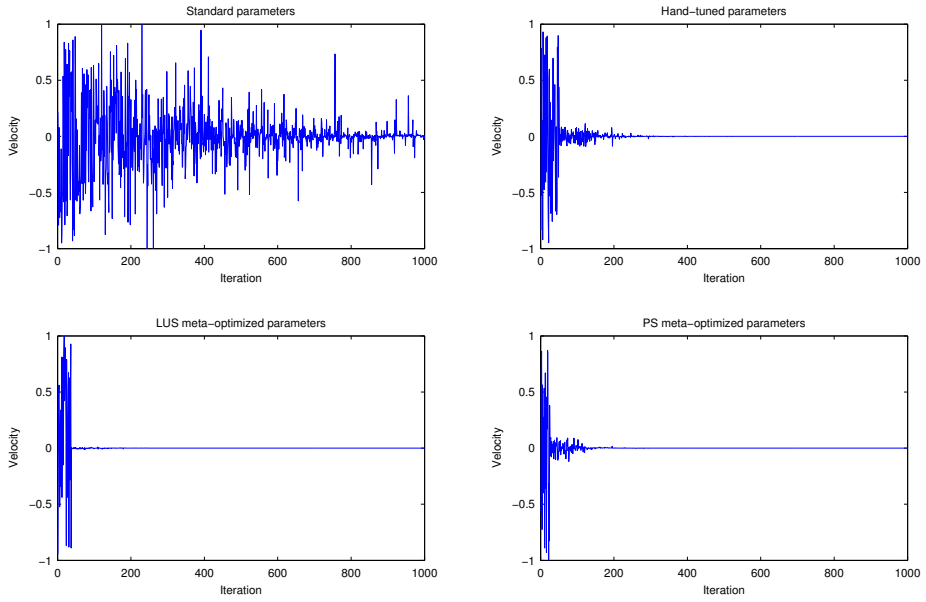


Figure 7.6: Velocity of one of the PSO particles using the standard [86, 101], hand-tuned, LUS meta-optimized and PS meta-optimized parameters.

A similar conclusion can be drawn when observing the position of the particle. In Figure 7.7, a similar behaviour is shown when using the standard [86, 101], hand-tuned, LUS meta-optimized and PS meta-optimized parameters. The position varies between -2 and 2 during the 1000 iterations when using the standard parameters. But when using LUS and PS meta-optimized parameters, the particle settles faster, before the 400 iterations.

After running the DE, SA, HC and PSO algorithms, a set of array weight vectors is obtained by each algorithm for the 16-element linear antenna array. These vectors are shown in Table 7.10. Note that the vectors have 32 elements because the array weights consist of a complex number that represents the amplitude and the phase of each antenna element.

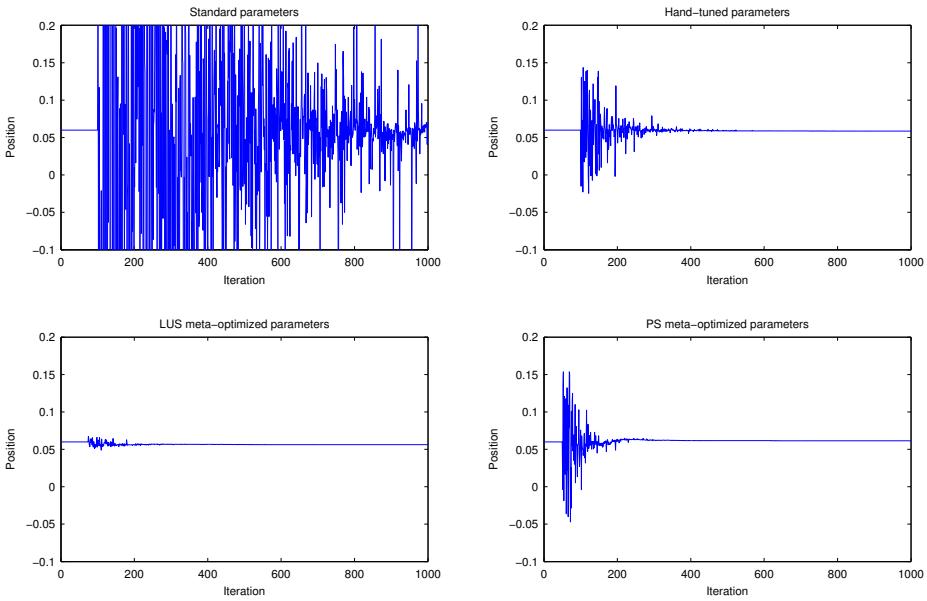


Figure 7.7: First position of one of the PSO particles using the standard [86], hand tuned, LUS meta-optimized and PS meta-optimized parameters.

	Standard
DE	0.50, -0.72,-1.00, 0.17,-0.28,-0.99, 0.12,-0.44,-0.98,-0.51,-0.77,-1.00,-0.64, 0.21, 0.22,-1.00,-0.65,-0.99,-1.00,-0.55,-1.00,-0.34,-0.88,-0.77,-0.63,-1.00,-0.84,-0.98,-0.77,-0.67,-0.08,-0.76
SA	0.87, 0.97, 0.59, 0.65, 0.60,-0.05, 0.39,-0.07,-0.21, 0.71, 0.48, 0.78, 0.13, 0.92, 0.58,-0.55,-0.56,-0.70,-0.33,-0.33, 0.09,-0.35, 0.15, 0.26,-0.44,-0.05, 0.23,-0.27, 0.16,-0.80,-0.69,-0.77
HC	0.88, -0.29, 0.36, 0.44, 0.23, 0.69, 0.60,-0.31, 0.52,-0.85, 0.71, 0.57,-0.38,-0.43,-0.49, 0.65, 0.34, 0.85, 0.96, 0.53, 0.90,-0.47,-0.62, 0.57, 1.00,-0.09, 0.29,-0.12, 0.78, 0.69,-0.31,-0.53
PSO	1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 0.79, 1.00, 1.00, 0.87, 0.57,-1.00,-1.00,-0.73, 0.16,-0.43,-1.00,-1.00,-1.00,-1.00,-0.95,-1.00,-1.00,-1.00,-1.00,-1.00,-1.00
	Hand-tuned
DE	0.47, 0.63, 0.54,-0.42, 0.44,-0.30, 0.79, 0.12,-0.20, 0.66, 0.22, 0.85, 0.82, 0.11, 0.68,-0.00, 0.58,-0.35,-0.47,-0.82,-0.89,-0.98,-0.61,-0.68,-0.47,-0.43, 0.63, 0.19, 0.42,-0.42, 0.76,-0.47
SA	0.47, -0.20,-0.76,-0.70,-0.66,-0.84,-0.59,-0.77,-0.80,-0.03,-0.29,-0.74,-0.53,-0.26,-0.42,-0.82, 0.57, 0.16, 0.84,-0.24,-0.46,-0.57,-0.29,-0.84, 0.14, 0.20, 0.90,-0.63,-0.23,-0.49, 0.46, 0.21
HC	-0.34, 0.93, 0.86, 0.74,-0.86,-0.41, 0.16, 0.36, 1.00,-0.43,-0.50, 0.16,-0.14,-0.55,-0.12, 0.11,-1.00, 0.00,-0.44,-0.30, 0.49,-1.00,-1.00,-0.69, 0.41,-0.95,-1.00,-0.76,-1.00,-1.00,-0.84,-1.00
PSO	0.05, -0.28,-0.08, 0.01, 0.33, 0.25, 0.19, 0.54,-0.23,-0.03, 0.31,-0.48, 0.07, 0.45, 0.21,-0.28, 0.62, 0.48, 0.07, 0.45, 0.02,-0.43, 0.13, 0.29, 0.11, 0.22, 0.74,-0.07, 0.26, 0.49, 0.63, 0.47
	PS meta-optimized
DE	-0.88,-0.97,-0.29,-0.77,-0.99,-0.90,-0.72,-0.99,-1.00,-0.33, 0.06,-1.00,-0.28,-0.33,-0.16,-0.04, 0.74, 1.00, 0.65, 0.72, 0.36,-0.43, 0.60, 1.00, 0.66, 0.33, 0.92, 0.96, 0.33,-0.00, 0.18, 0.26
SA	-0.75,-0.20,-0.62,-1.00,-0.94,-0.85,-0.78,-0.16,-0.40,-0.14,-0.79, 0.15,-0.68,-0.97,-0.42,-1.00, 0.94, 0.04, 0.69, 0.42, 0.71, 0.43,-0.07, 0.42, 0.33,-0.04, 0.63,-0.85,-0.19,-0.07, 0.37, 0.54
HC	-0.16, 0.27, 0.07,-0.37,-0.65, 0.57,-0.02, 0.32, 0.73,-0.11,-0.73, 0.69, 0.40, 0.52,-0.56,-0.01,-0.68, 0.19, 0.08,-0.79,-0.39,-0.62,-0.60,-0.65,-0.97,-0.56,-0.74,-0.85,-0.65,-0.72,-0.78,-0.96
PSO	0.46, 0.15,-0.79,-0.04,-0.56, 0.24, 0.13,-0.43, 0.18, 0.08,-0.27, 0.42,-0.29, 0.33,-0.10, 0.23, 1.00, 0.78, 1.00, 0.59, 0.61, 1.00, 0.65, 0.85, 0.94, 0.81, 0.99, 0.90,-0.48, 0.80, 0.04, 0.22
	LUS meta-optimized
DE	-0.81,-0.68, 0.11,-0.12,-0.00,-0.81,-0.40,-0.94,-0.08,-0.41,-0.12,-0.47,-0.28,-0.75,-0.12,-0.77,-0.98,-0.36,-0.58,-0.70,-0.54,-0.95,-1.00,-0.60,-0.90,-0.92,-0.86,-0.31,-0.40,-0.98,-0.73,-0.97
SA	-0.11, 0.64, 0.10, 0.86, 1.00, 0.79, 0.90, 0.90, 0.94, 0.57, 0.30, 0.24,-1.00, 0.15, 1.00, 0.96,-0.34,-1.00,-0.77, 0.28, 0.73,-0.39, 0.01, 0.08,-0.24,-0.03,-0.21, 0.13,-0.71,-0.24,-0.36,-0.74
HC	0.40, -0.25,-1.00,-0.76,-0.62,-1.00,-0.28,-1.00,-1.00,-0.64,-1.00,-1.00,-0.46, 1.00,-0.55,-1.00,-1.00, 1.00, 1.00, 0.22, 0.36,-1.00,-1.00,-0.53,-1.00,-1.00,-1.00,-0.41,-1.00,-1.00,-0.31
PSO	1.00, 0.26, 1.00, 1.00, 1.00, 0.42, 0.56, 1.00, 0.22, 0.88, 0.94, 1.00, 0.65, 1.00, 0.50, 1.00, 1.00, 0.91, 1.00, 0.94, 0.99, 0.90,-0.22, 0.45, 1.00, 1.00, 0.77, 1.00, 1.00, 0.47, 1.00, 1.00
	DE meta-optimized
DE	-0.70,-0.99,-1.00,-0.78,-0.65,-1.00,-1.00,-0.99,-0.98,-0.30, 0.34, 0.24,-0.40,-1.00,-0.54,-0.99, 1.00, 0.63, 1.00, 1.00, 0.92, 1.00, 0.77, 1.00, 0.68, 0.22, 0.99, 0.59, 0.83, 0.54,-0.08, 0.34
SA	0.06, -0.13,-0.89,-0.66,-0.81,-0.40,-0.49, 0.19,-0.72,-0.27, 0.28,-0.24,-0.50, 0.19,-0.84,-0.60,-0.34,-0.63,-0.80,-0.72,-0.83,-0.76, 0.04,-0.51,-0.91,-0.10,-0.08,-0.45, 0.24,-0.10,-0.31, 0.10
HC	1.00, -0.57,-0.63,-0.30,-0.12, 1.00,-1.00,-0.67,-1.00,-1.00,-1.00,-1.00,-1.00,-0.39,-1.00,-0.65,-1.00,-0.97,-1.00,-1.00, 1.00,-1.00,-1.00,-1.00,-0.37,-1.00, 1.00,-1.00,-0.79,-0.13,-1.00, 0.47
PSO	0.02, 1.00, 0.29, 0.72, 0.02, 0.99, 0.84, 0.17, 0.43, 0.90, 0.38, 0.07, 0.97, 1.00, 0.84, 1.00,-0.54,-0.95,-0.50,-0.36,-0.73, 0.08, 0.92,-0.96,-0.75,-0.67,-0.71,-0.03,-0.96,-0.61,-0.75, 0.08
	PSO meta-optimized
DE	-0.11,-0.17, 0.65,-0.40,-0.12,-0.20, 0.25,-0.50, 0.36, 0.36, 0.06,-0.71, 0.36,-0.32, 0.09,-0.08, 0.38, 0.76, 1.00, 1.00, 0.56, 0.51, 0.48, 0.76, 0.61, 0.60, 0.88, 0.89, 0.45, 0.81, 0.99, 0.62
SA	0.81, 0.38, 0.31, 0.36, 0.79, 0.18, 0.83, 0.49, 0.69, 0.78, 0.36, 0.57, 0.76, 0.97, 0.60, 0.15,-0.08, 0.11,-0.74,-0.81,-0.84, 0.08, 0.15, 0.23,-0.29,-0.31,-0.55, 0.01,-0.40,-0.22, 0.24,-0.70
HC	-1.00, 0.10,-0.79, 0.26,-1.00,-1.00, 0.42, 0.21, 0.45, 0.70,-1.00,-0.15,-1.00,-1.00,-0.53,-1.00,-1.00,-0.11,-1.00,-0.33,-0.43,-0.51,-1.00, 0.41, 0.10,-0.16,-1.00, 1.00,-1.00,-0.10, 0.36,-1.00
PSO	0.81, 0.38, 0.62, 0.63, 0.29, 0.48, 0.20, 0.60, 0.13, 0.41, 0.40, 0.04, 0.21, 0.54, 0.57, 0.25, 0.47, 0.08,-0.56,-0.45,-0.63,-0.05,-0.02,-0.17,-0.33,-0.77, 0.37,-0.06,-0.14,-0.20,-0.15, 0.08

Table 7.10: Antenna weights vectors obtained with the DE, SA, HC and PSO algorithms using the meta-optimized parameters calculated by the PS, LUS, DE and PSO meta-optimizers.

7.4.4 Statistical Analysis

During the optimization process, statistical data of the performance of the DE, SA, HC and PSO algorithms was collected. As stated previously, the algorithms were run 100 times each to obtain statistical significance and it is the mean, standard deviation, lower and upper quartiles, minimum, maximum and sum that Table 7.11 shows. It can be noticed that when using the standard parameters, particularly for the SA and HC algorithms, the mean fitness is much higher (0.60 and 0.63) compared with the (0.007 and 0.002) of the DE and PSO algorithms. Consequently, the sum values are higher as can be seen at the end of the table.

The statistical data is shown in graphical form in the following figures. Each figure shows the results for the DE, SA, HC and PSO methods, and each one has been optimized with one of the four meta-optimizers (PS, LUS, DE and PSO). The figure consists of a graph with the mean of 100 runs of each fitness function over the number of iterations. It also presents three box plots corresponding to the standard, hand-tuned and meta-optimized results. The box plot was found to be a convenient way of graphically depicting the fitness function. In this case, the box plot has the following features:

- The top and bottom of each box are the Q1 and Q3 which are the 25th and 75th percentiles of the samples, respectively.
- The whiskers are lines extending above and below each box. Whiskers are drawn from the ends of the interquartile ranges to the furthest observations within the whisker length.
- Observations beyond the whisker length are marked with a red + sign and are known as outliers. In this study, an outlier is a value that is more than 1.5 times the interquartile range away from the top or bottom of the box.

Figures 7.8, 7.9, 7.10 and 7.11 show the results for the Differential Evolution method when using parameters meta-optimized with the LUS algorithm with 200 iterations. It can be observed that the hand-tuned parameters obtain a better mean fitness function value compared to the standard parameters. Moreover, the meta-optimized parameters achieve lower values still. On the other hand it can

Parameters	Method	Mean	Std.Dev.	Q1	Q3	Min	Max	Sum
Standard	DE	0.0076	0.0043	0.0047	0.0050	0.0015	0.0267	0.7586
	SA	0.6081	0.2179	0.5834	0.7417	0.1049	0.9607	60.811
	HC	0.6374	0.2330	0.9456	0.7652	0.1280	0.9741	63.738
	PSO	0.0021	0.0008	0.0020	0.0030	0.0009	0.0051	0.2147
Hand-tuned	DE	0.0496	0.0218	0.0820	0.0312	0.0095	0.1128	4.9628
	SA	0.0109	0.0045	0.0117	0.0089	0.0039	0.0246	1.0948
	HC	0.0899	0.0327	0.0668	0.1182	0.0214	0.1831	8.9945
	PSO	0.0087	0.0103	0.0041	0.0019	0.0019	0.0819	0.8654
PS meta-optimized	DE	0.0018	0.0005	0.0022	0.0013	0.0009	0.0028	0.1785
	SA	0.0022	0.0006	0.0016	0.0019	0.0011	0.0052	0.2176
	HC	0.0026	0.0008	0.0014	0.0033	0.0013	0.0047	0.2580
	PSO	0.0024	0.0010	0.0021	0.0025	0.0010	0.0069	0.2378
LUS meta-optimized	DE	0.0016	0.0005	0.0010	0.0012	0.0008	0.0032	0.1591
	SA	0.0034	0.0011	0.0040	0.0039	0.0020	0.0092	0.3441
	HC	0.0862	0.0302	0.0551	0.0606	0.0091	0.1809	8.6174
	PSO	0.0024	0.0011	0.0022	0.0013	0.0007	0.0084	0.2365
DE meta-optimized	DE	0.0017	0.0005	0.0020	0.0015	0.0008	0.0036	0.1703
	SA	0.0026	0.0009	0.0017	0.0028	0.0013	0.0067	0.2578
	HC	0.0903	0.0306	0.0833	0.1437	0.0231	0.1628	9.0349
	PSO	0.0024	0.0011	0.0011	0.0018	0.0010	0.0066	0.2374
PSO meta-optimized	DE	0.0017	0.0005	0.0011	0.0013	0.0009	0.0033	0.1664
	SA	0.0024	0.0007	0.0018	0.0026	0.0010	0.0057	0.2366
	HC	0.0885	0.0300	0.1238	0.0575	0.0147	0.1540	8.8451
	PSO	0.0025	0.0010	0.0017	0.0036	0.0011	0.0062	0.2469

Table 7.11: Statistics of 100 runs of the DE, SA, HC and PSO algorithms when using the standard and hand-tuned parameters as well as those obtained by meta-optimization using the PS, LUS, DE and PSO algorithms. The problem to be solved was a 16-element linear antenna array with a main lobe at 0° and nulls at 30° , 60° and -45°.

also be seen in the box plots that there is a significant variation in each of the 100 runs when using the standard parameters. The hand-tuned box plot shows better progress and more regularity in the quartiles compared with the standard results. The meta-optimized box plot looks similar to the hand-tuned one, but a closer look shows that there is even less variance in the data after 120 iterations. Lastly it can be seen that the four meta-optimizers performed very similar to each other as can be seen on noted by comparing the four figures.

The results obtained when running the Simulated Annealing over 100 times are shown in Figures 7.12, 7.13, 7.14 and 7.15 for meta-optimizers PS, LUS, DE and PSO respectively. The fitness function graph shows similar results to those for the DE algorithm in that the mean values for the meta-optimized results are better compared to those obtained with the standard and hand-tuned parameters. It can also be observed in the box plots that there is a great variation in the different runs when using the standard parameters. The hand-tuned results show much better progress in terms of variation and a desired decrease in the fitness function values with every iteration. The box plot for the meta-optimized parameters shows more regularity as depicted in the quartiles, especially after 120 iterations when there are almost no outliers (marked with a red + sign), unlike the hand-tuned results. Finally, as observed previously, there is no great difference between meta-optimizers. The PS, LUS, DE and PSO algorithms were able to meta-optimize the problems in a similar manner.

A different situation can be observed in Figures 7.16, 7.17, 7.18 and 7.19 where the Hill Climb algorithm was tested. The mean fitness function graph shows little difference between the fitness values obtained by the algorithm using hand-tuned parameters and with meta-optimized ones. Although the exception is when using the PS meta-optimizer as can be seen in Figure 7.17. One possible explanation for this is that the number of meta-iterations in the particular case of the Hill Climb algorithms was set to 40 as shown previously in Table 7.3. As explained before, the number of meta-optimizations was decided according to the total number of parameters to be meta-optimized. In the case of HC, there are only two parameters r and D, so the meta-optimizers performed only 40 meta-iterations compared to 60 and 80 in the case of the DE, SA and PSO

algorithms.

Finally, the Particle Swarm Optimization was also run for 100 times and the results are shown in Figures 7.20, 7.21, 7.22 and 7.23. It can be observed that the fitness values obtained with the meta-optimized parameters are better compared to those produced by the standard and hand-tuned ones. Although, unlike the case when meta-optimizing the Simulated Annealing algorithm, the values from the standard PSO parameters show some grade of consistency. This is due to the extensive research [86, 101] that has been carried out to obtain these parameters. Although the hand-tuned parameters calculated in this thesis are able to obtain more regularity between different runs as can be seen in the box plots, this demonstrates that the selection of parameters when using optimization techniques depends greatly on the particular problem that is being solved.

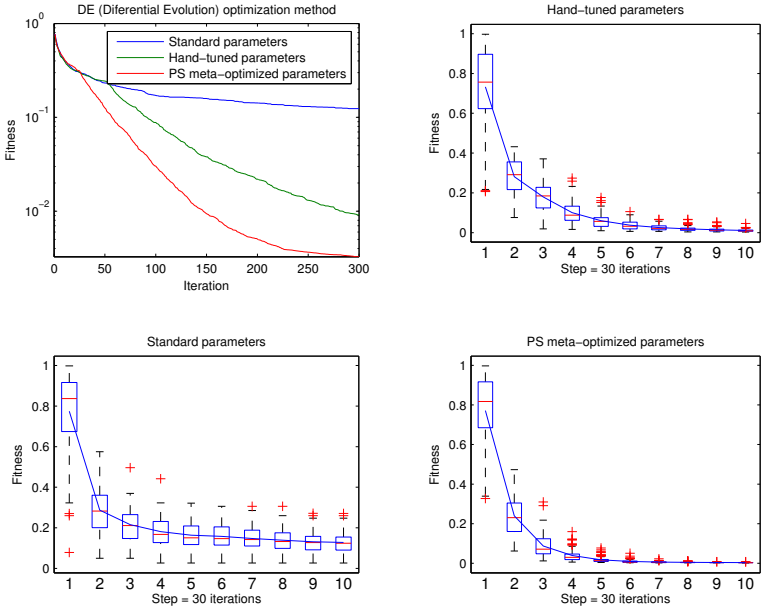


Figure 7.8: Mean fitness after 100 runs of the DE method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PS for 200 iterations.

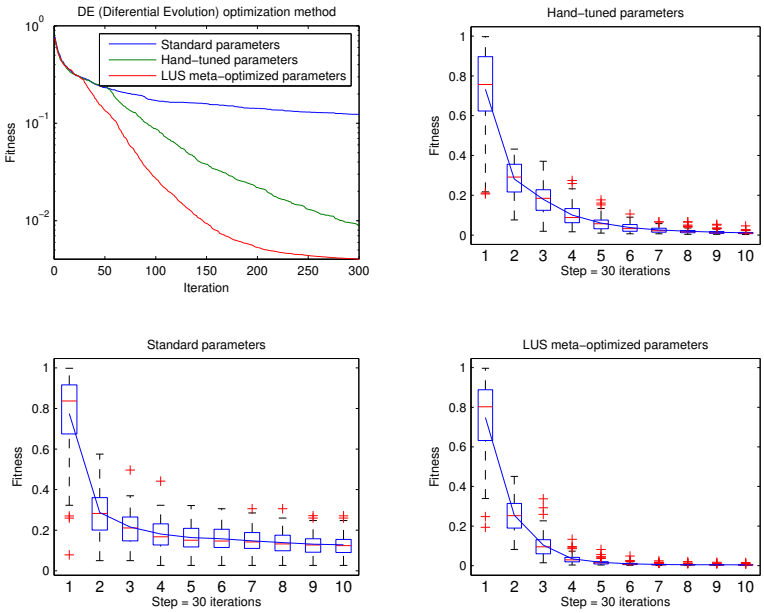


Figure 7.9: Mean fitness after 100 runs of the DE method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with LUS for 200 iterations.

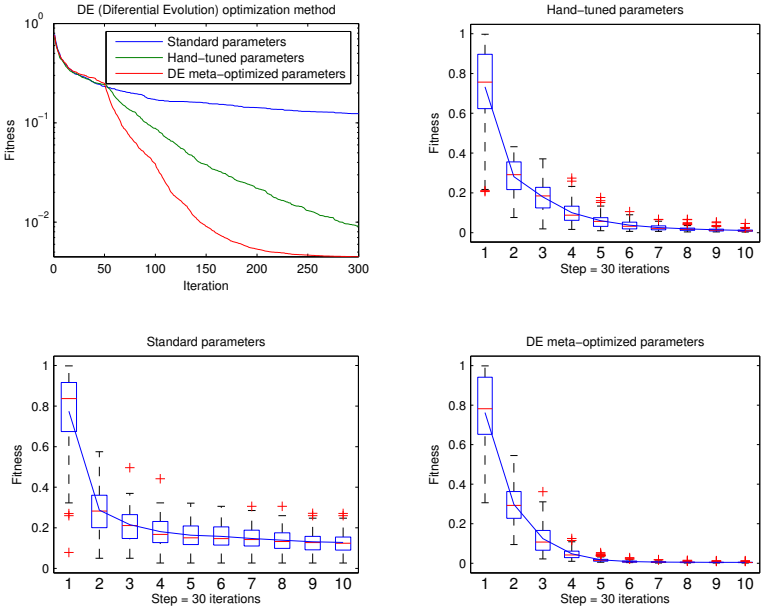


Figure 7.10: Mean fitness after 100 runs of the DE method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with DE for 200 iterations.

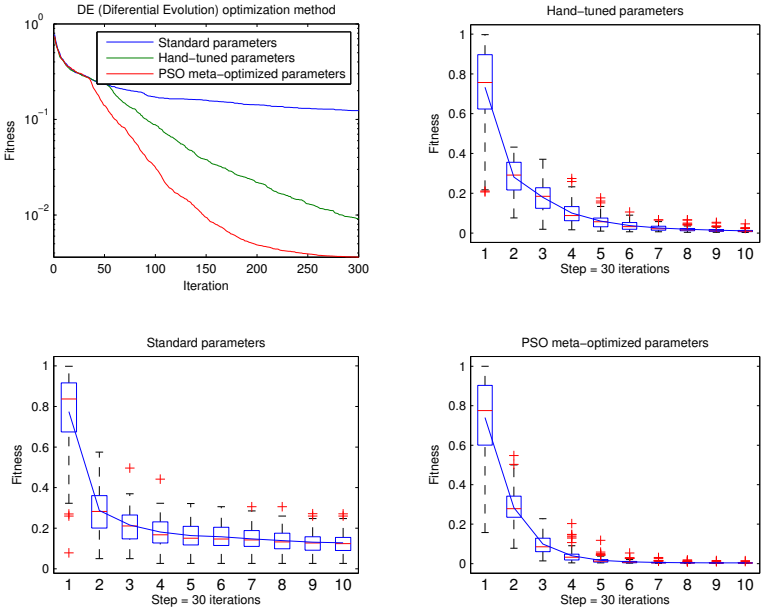


Figure 7.11: Mean fitness after 100 runs of the DE method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PSO for 200 iterations.

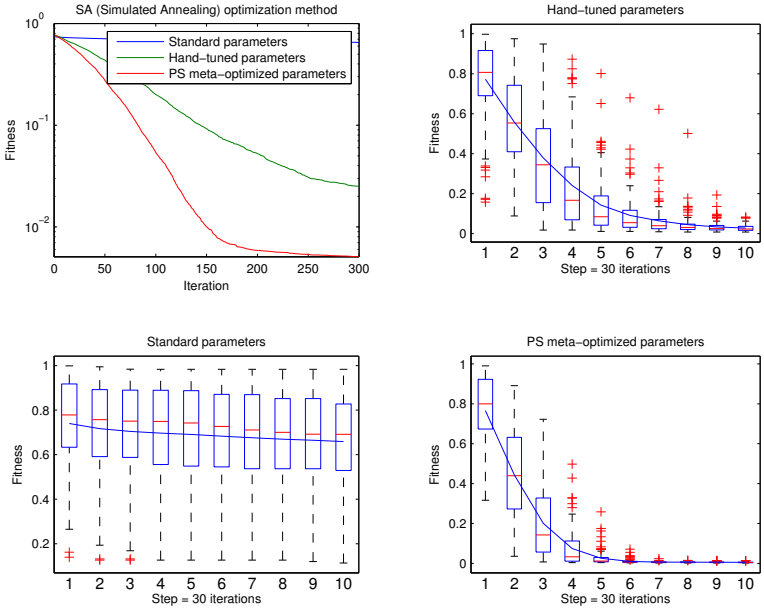


Figure 7.12: Mean fitness after 100 runs of the SA method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PS for 200 iterations.

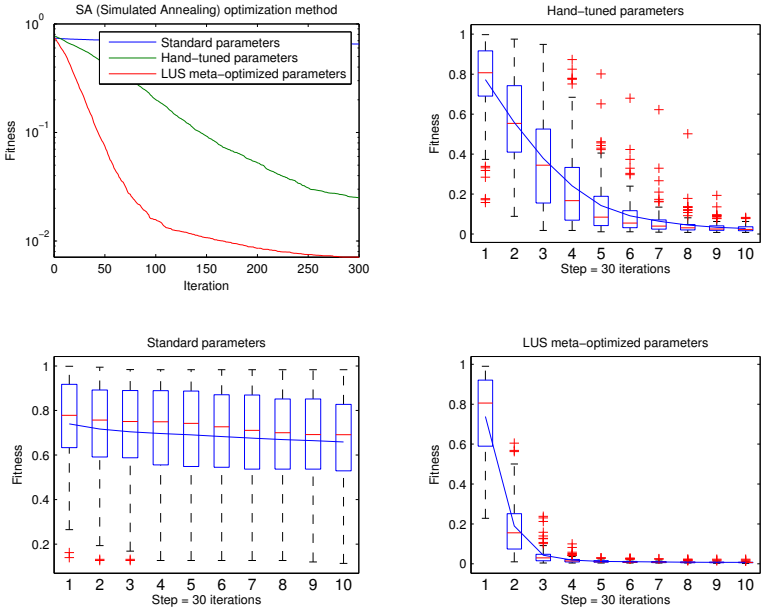


Figure 7.13: Mean fitness after 100 runs of the SA method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with LUS for 200 iterations.

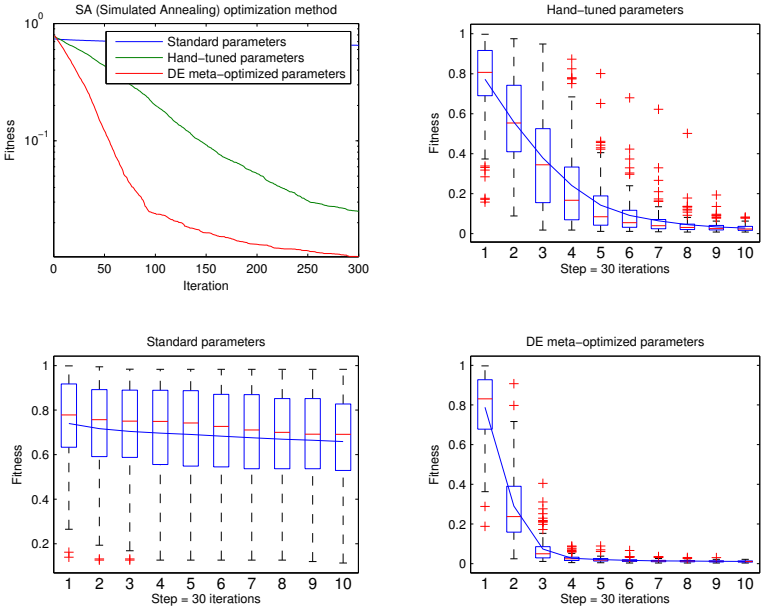


Figure 7.14: Mean fitness after 100 runs of the SA method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with DE for 200 iterations.

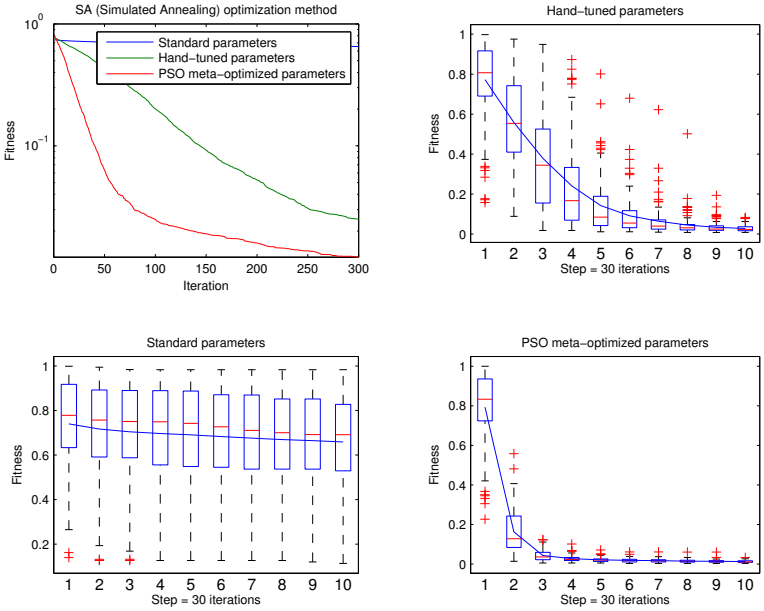


Figure 7.15: Mean fitness after 100 runs of the SA method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PSO for 200 iterations.

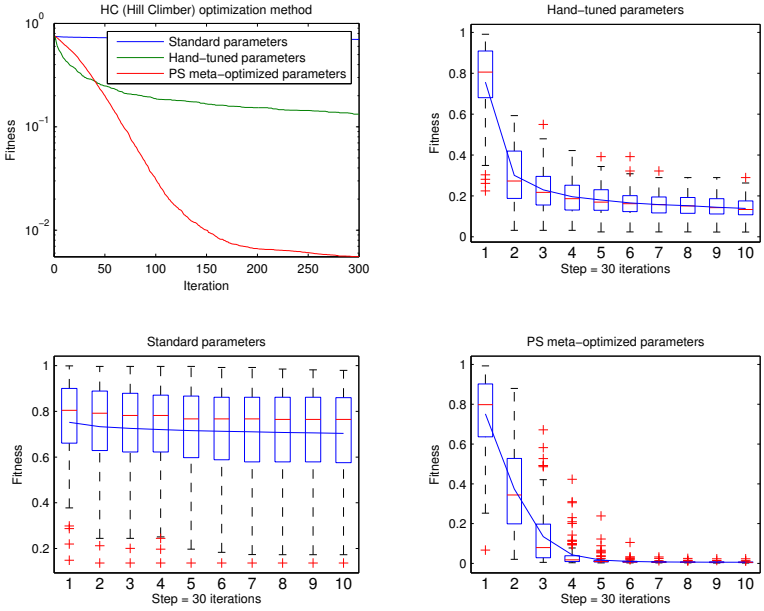


Figure 7.16: Mean fitness after 100 runs of the HC method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PS for 200 iterations.

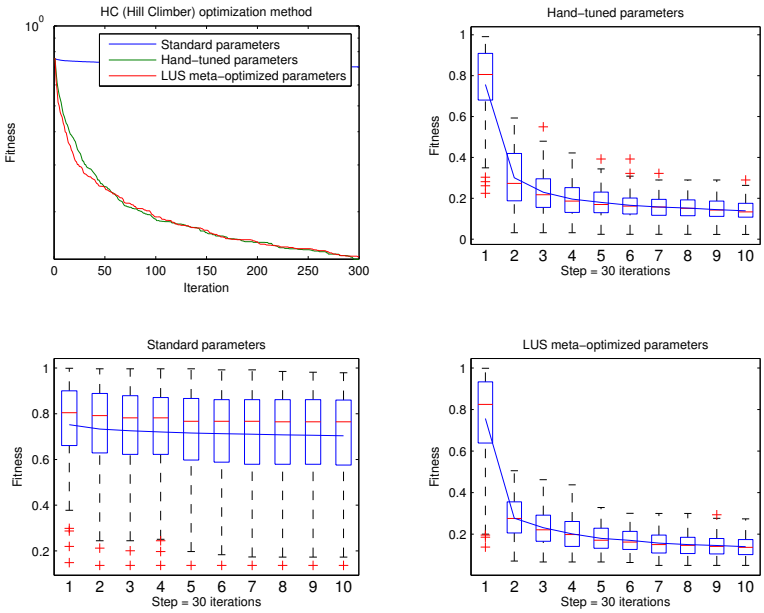


Figure 7.17: Mean fitness after 100 runs of the HC method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with LUS for 200 iterations.

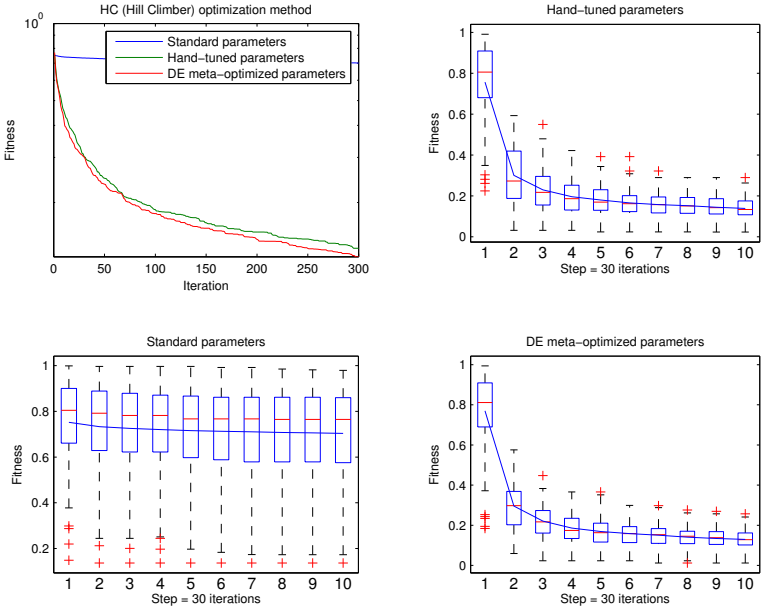


Figure 7.18: Mean fitness after 100 runs of the HC method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with DE for 200 iterations.

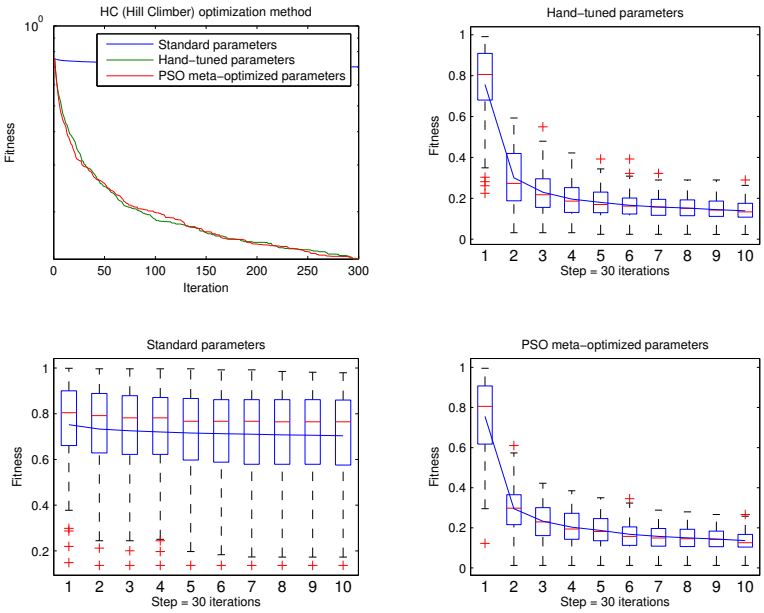


Figure 7.19: Mean fitness after 100 runs of the HC method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PSO for 200 iterations.

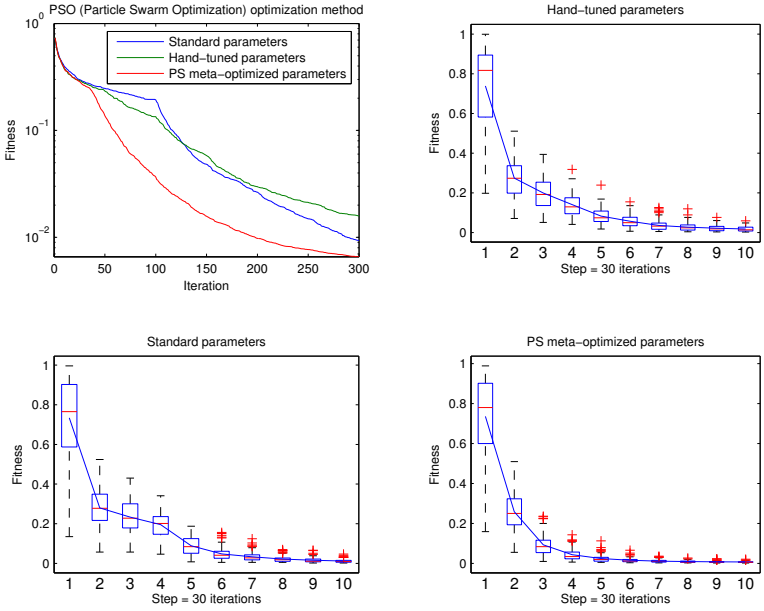


Figure 7.20: Mean fitness after 100 runs of the PSO method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PS for 200 iterations.

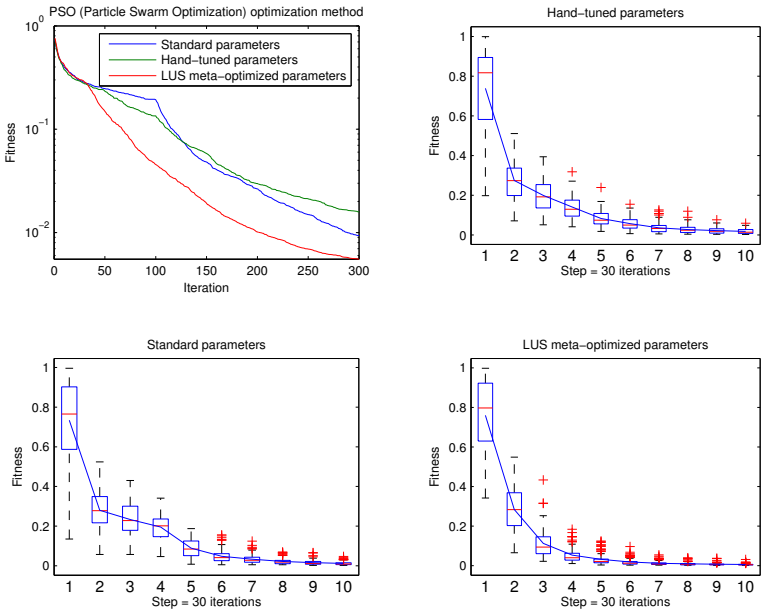


Figure 7.21: Mean fitness after 100 runs of the PSO method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with LUS for 200 iterations.

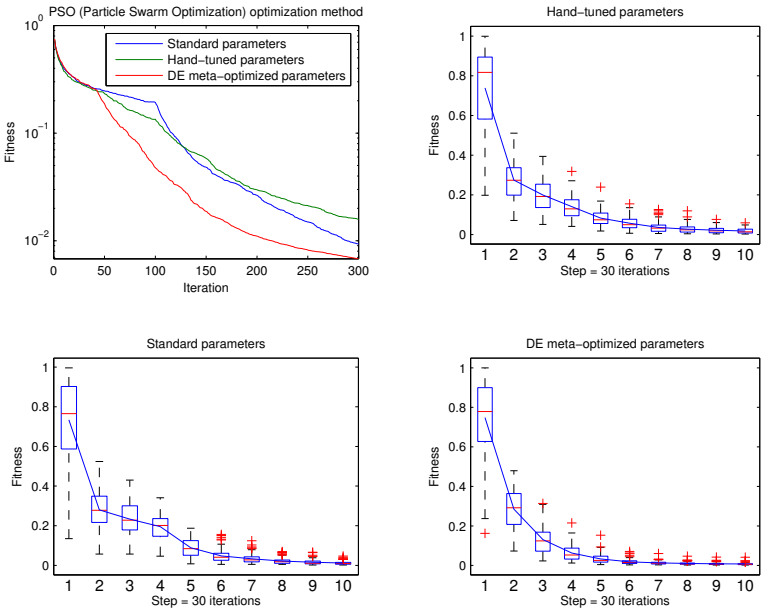


Figure 7.22: Mean fitness after 100 runs of the PSO method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with DE for 200 iterations.

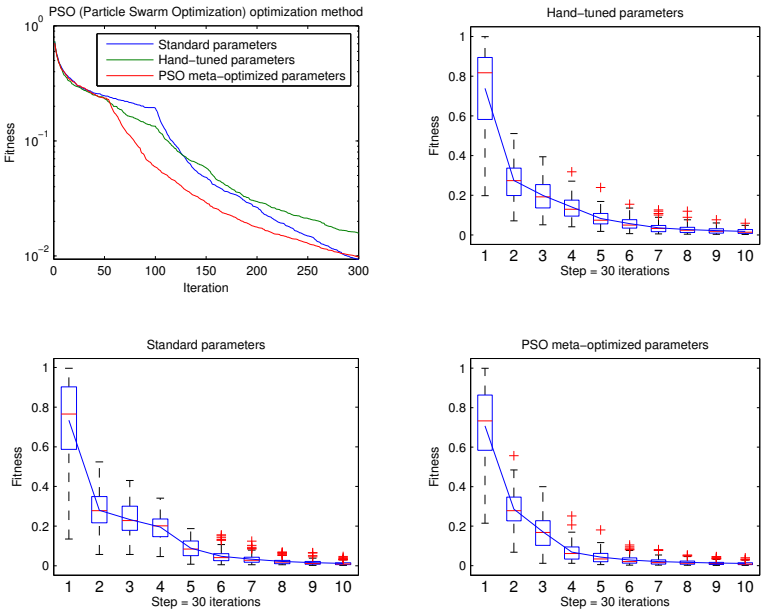


Figure 7.23: Mean fitness after 100 runs of the PSO method and quartiles using standard, hand-tuned and meta-optimized parameters. Meta-optimization with PSO for 200 iterations.

7.5 Geometry Synthesis and Meta-optimization

In Chapter 4, the importance of the spatial position of antenna elements in an antenna array was stressed. It was explained that the radiation pattern was affected by displacing the antenna elements in such a way that it was possible to reduce sidelobe levels while preserving the gain. It is also possible to place nulls at the direction of undesired signals, thus reducing interference. In this section, meta-optimization techniques are employed to synthesis the antenna array geometry in order to obtain minimum sidelobe levels and null control.

The methods used for antenna array synthesis can be classified into two categories: deterministic and stochastic. The deterministic methods include analytical methods and semi-analytical methods. The deterministic methods in general are computationally time consuming as the number of elements in the array increases [102]. On the other hand, stochastic methods are nowadays very commonly used in electromagnetics [29, 77, 103]. Stochastic methods have the ability to deal with large number of optimization parameters, escaping from local minima and are easy to implement. Amongst others, the PSO algorithm has been extensively used for antenna synthesis in recent years [78].

In particular, different variations of the PSO have been employed in the area of antenna synthesis. In [15], Khodier formulated a fitness function used by the PSO to obtain minimum sidelobe levels and null control by calculating the area under the curve of the desired array pattern. The distance between antenna elements of a linear antenna array was optimized and the results were compared with the QPM (Quadratic Programming Method) technique. This concept was later used in [104] where the synthesis was carried out for planar arrays and in [83] for circular arrays. Other similar work is presented in [105] where different evolutionary optimization techniques were used to reduce sidelobe levels of circular arrays. In addition, modifications to the PSO algorithm were proposed, namely the NPSO in [106] and IPSO in [107].

In this thesis, the approach proposed by Khodier in [15] has been studied and the meta-optimization techniques explained in this chapter were applied to the same antenna problem. This was done in order to compare the performance of the PSO algorithm when using meta-optimized parameters to obtain desired

antenna geometry which presents minimum sidelobe levels as well as null control. The problem to be solved is the second example in [15] which consists of a 28-element linear array designed for SLL suppression in the region [0°, 180°] and prescribed nulls at 55°, 57.5°, 50°, 120°, 122.5°and 125°. To achieve this, the following function is used to evaluate the fitness:

$$Fitness = \sum_{i} \frac{1}{\Delta\phi_i} \int_{\phi_{li}}^{\phi_{ui}} |AF(\phi)|^2 \, d\phi + \sum_{k} |AF(\phi_k)|^2 \tag{7.1}$$

where $[\phi_{li}, \phi_{ui}]$ are the spatial regions in which the SLL is suppressed, in this case from 0° to 180°. $\Delta\phi_i = \phi_{ui} - \phi_{li}$, and ϕ_k are the directions of the nulls.

After the meta-optimization process, the parameters found for the PSO are: $NP = 30$, $\omega = 0.130108$, $c_1 = 0.470517$ and $c_2 = 1.846860$. These parameters are then used by the PSO to optimize the antenna synthesis problem. The results of the optimization are shown in Table 7.12 and are the vectors corresponding to the distances of each antenna element to the array centre. The resulting antenna pattern is shown in Figure 7.24. It can be noted that lower sidelobe levels can be achieved when using the meta-optimized parameters.

7.6 Summary

In this chapter, a technique called meta-optimization was investigated. Meta-optimization consists of employing a second optimization algorithm to find good behavioural parameters for a given technique. The meta-landscapes for different optimization schemes were presented and it was concluded that the problem of finding better parameters was relatively simple when using non-complex algorithms like PS and LUS. Experiments were carried out in which an antenna problem was solved by using standard, hand-tuned and meta-optimized parameters. The results showed a better progress in terms of the number of iterations as well as consistency in the values of fitness functions. Lastly, an antenna synthesis problem was solved using meta-optimized parameters and, as shown in Figure 7.24, it was found that the PSO algorithm achieved lower sidelobe levels compared to those presented in previous publications.

Standard														
DE	±4.5180,	±8.6862,	±10.8436,	±9.9843,	±11.9996,	±0.4812,	±7.9438,	±6.3044,	±1.2358,	±3.0473,	±2.3479,	±3.2755,	±13.3881,	±4.8124
SA	±0.5125,	±1.3939,	±2.4473,	±3.3485,	±4.3952,	±5.1976,	±6.2690,	±7.3942,	±8.3615,	±9.3845,	±10.6289,	±11.7225,	±12.7543,	±14.0000
HC	±0.4431,	±1.4721,	±2.4617,	±3.6213,	±4.6020,	±5.5566,	±6.6317,	±7.5646,	±8.5405,	±9.5126,	±10.5356,	±11.5328,	±12.4722,	±13.6365
PSO	±0.4951,	±1.4006,	±2.4292,	±3.4545,	±4.4973,	±5.4815,	±6.5087,	±7.5120,	±8.4493,	±9.3629,	±10.6127,	±11.6405,	±12.6884,	±14.0000
Hand-tuned														
DE	±3.2202,	±0.1324,	±5.2291,	±7.4431,	±6.1098,	±14.0000,	±11.3948,	±10.6535,	±9.2418,	±1.2105,	±12.9482,	±4.1855,	±1.8324,	±12.1138
SA	±0.5000,	±1.5000,	±2.5000,	±3.5000,	±4.5000,	±5.5000,	±6.5000,	±7.5000,	±8.5000,	±9.5000,	±10.5000,	±11.5000,	±12.5000,	±13.5000
HC	±0.4431,	±1.4721,	±2.4617,	±3.6213,	±4.6020,	±5.5566,	±6.6317,	±7.5646,	±8.5405,	±9.5126,	±10.5356,	±11.5328,	±12.4722,	±13.6365
PSO	±10.1007,	±0.4897,	±7.2276,	±12.6023,	±8.8775,	±11.3336,	±2.3531,	±5.2204,	±8.1791,	±1.3215,	±13.9800,	±6.2984,	±3.3035,	±4.4046
PS meta-optimized														
DE	±0.5102,	±1.5006,	±2.4958,	±3.4568,	±4.5287,	±5.4952,	±6.8577,	±8.0736,	±8.9429,	±9.9774,	±10.8999,	±11.7458,	±12.7162,	±14.0000
SA	±0.5207,	±1.4770,	±2.4679,	±3.4782,	±4.4888,	±5.4270,	±6.5193,	±7.5350,	±8.4485,	±9.3813,	±10.6006,	±11.6179,	±12.6921,	±14.0000
HC	±0.5239,	±1.4809,	±2.4961,	±3.5229,	±4.5352,	±5.4557,	±6.4973,	±7.4349,	±8.3700,	±9.2835,	±10.5623,	±11.6479,	±12.7008,	±14.0000
PSO	±3.0349,	±6.4483,	±13.9999,	±0.6313,	±12.2137,	±9.7964,	±12.8550,	±1.7781,	±7.6411,	±11.3429,	±10.7613,	±5.2548,	±4.2565,	±8.7276
LUS meta-optimized														
DE	±0.5643,	±1.6174,	±2.6328,	±3.5783,	±4.5823,	±5.5469,	±6.6975,	±7.7340,	±8.6096,	±9.4410,	±10.5314,	±11.5534,	±12.6238,	±14.0000
SA	±0.6083,	±1.4877,	±2.4544,	±3.5863,	±4.5218,	±5.5397,	±6.6174,	±7.5637,	±8.4352,	±9.4371,	±10.5745,	±11.4719,	±12.5034,	±13.7681
HC	±0.4617,	±1.3924,	±2.3920,	±3.4149,	±4.4666,	±5.3847,	±6.4326,	±7.4821,	±8.4630,	±9.4701,	±10.7099,	±11.6817,	±12.6989,	±14.0000
PSO	±0.7349,	±5.3588,	±13.9958,	±11.0294,	±7.7529,	±6.5539,	±12.7723,	±10.2245,	±1.8965,	±8.8950,	±3.7413,	±11.9684,	±4.6704,	±2.8647
DE meta-optimized														
DE	±0.5228,	±1.3753,	±2.3951,	±3.3743,	±4.4098,	±5.3170,	±6.4403,	±7.5772,	±8.4904,	±9.4348,	±10.5898,	±11.5583,	±12.6690,	±14.0000
SA	±0.5501,	±1.4681,	±2.5115,	±3.5286,	±4.5478,	±5.4476,	±6.4568,	±7.4053,	±8.3436,	±9.5113,	±10.6259,	±11.5256,	±12.4320,	±13.6036
HC	±0.4972,	±1.5003,	±2.5166,	±3.5031,	±4.5233,	±5.4554,	±6.5532,	±7.5856,	±8.4824,	±9.3976,	±10.5749,	±11.6072,	±12.6849,	±14.0000
PSO	±0.5167,	±1.4307,	±2.4098,	±3.4446,	±4.4675,	±5.4957,	±6.6504,	±7.6789,	±8.5943,	±9.5119,	±10.6849,	±11.6707,	±12.7225,	±14.0000
PSO meta-optimized														
DE	±5.3860,	±1.5919,	±10.2384,	±0.5966,	±3.5194,	±7.7448,	±4.5140,	±11.1483,	±14.0000,	±12.0322,	±2.5842,	±12.8350,	±8.9127,	±6.5673
SA	±0.4605,	±1.2892,	±2.2539,	±3.2922,	±4.3483,	±5.2225,	±6.3363,	±7.4067,	±8.4003,	±9.3867,	±10.5849,	±11.6785,	±12.7158,	±14.0000
HC	±0.4783,	±1.4588,	±2.3847,	±3.5166,	±4.4009,	±5.4365,	±6.6059,	±7.5408,	±8.4656,	±9.4828,	±10.5085,	±11.4388,	±12.3847,	±13.5117
PSO	±0.5287,	±1.5553,	±2.6336,	±3.6876,	±4.6256,	±5.4840,	±6.5549,	±7.5746,	±8.4766,	±9.3269,	±10.4962,	±11.5102,	±12.6548,	±14.0000

Table 7.12: 16-element linear antenna array geometry obtained with the DE, SA, HC and PSO algorithms using the meta-optimized parameters calculated by the PS, LUS, DE and PSO meta-optimizers. Each value corresponds to the distance of each element to the centre of the array with respect to λ.

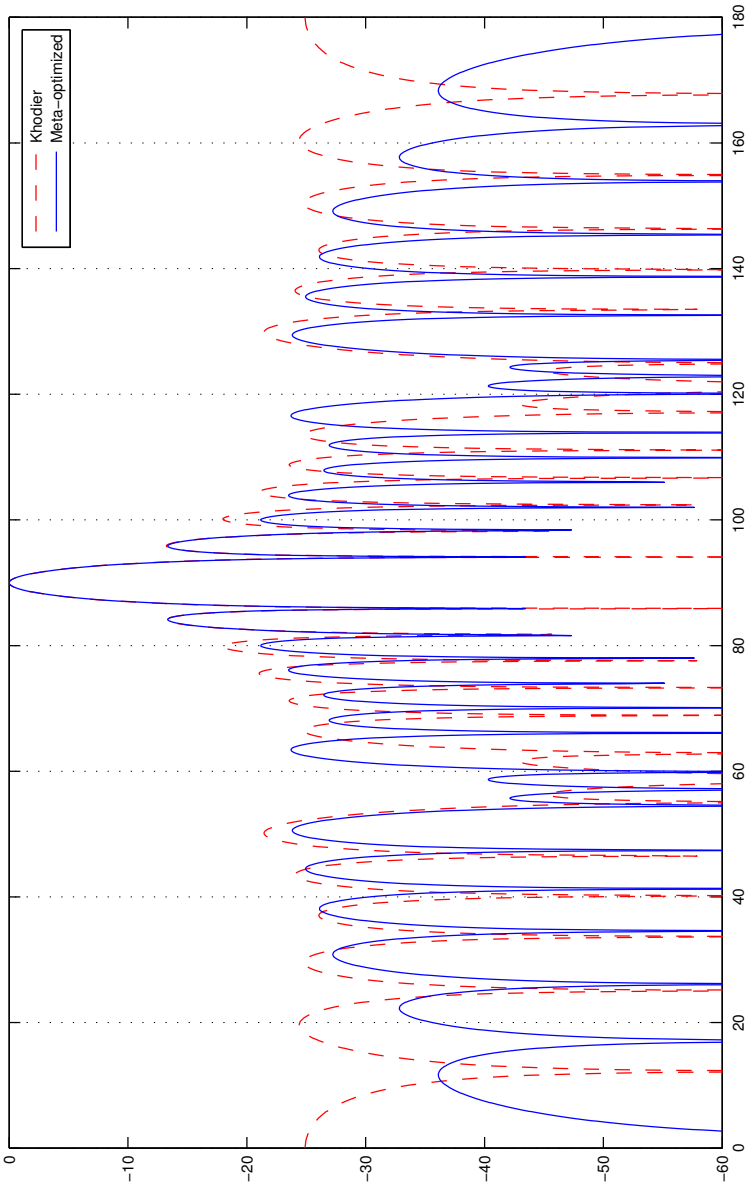


Figure 7.24: Lower sidelobe levels obtained with meta-optimized parameters compared with Khodier's results [15]. A 28-element array with SLL suppression region of [0°, 180°] and prescribed nulls at 55°, 57.5°, 60°, 120°, 122.5° and 125°.

Chapter 8

Summary and Conclusions

8.1 Introduction

The focus of this thesis has been to investigate the effectiveness of bio-inspired optimization algorithms for controlling smart antennas. This investigation was carried out by analysing, testing and comparing the different antenna array geometries. Measurements of directivity, half-power beamwidth and sidelobe levels as well as frequency response were obtained. The feasibility of using bio-inspired algorithms such as Particle Swarm Optimization and Genetic Algorithms to obtain an optimal antenna radiation pattern for a given problem was studied. The process of digitally shifting the phase weights was investigated as well as different approaches to computing the appropriate fitness function. A comparison of different optimization algorithms was carried out in terms of fitness function evaluations. Moreover, study of the impact of initial parameters on the ability of optimization algorithms applied to smart antennas was presented. This has paved the way to establish the best configuration parameters to enhance the efficiency and efficacy of bio-inspired algorithms for adaptive arrays.

The following sections present a summary of the work carried out in this thesis followed by the conclusions derived from the research presented herein. Finally, guidelines for future work are provided.

8.2 Summary

In recent years, wireless communications technologies have grown in such a manner that new adaptive techniques are required to reduce the effects of interference. The use of smart antennas in mobile devices can help eliminate co-channel interference and multi-access interference among other problems. These kind of antennas are able to radiate power towards a desired direction and simultaneously avoid interference by reducing power in other directions.

One way to control the antenna radiation pattern is to modify the antenna geometry in such a way that a desired beam is created. The design of antennas benefits from modern simulation software tools that allow the exploration of different antenna configurations. In Chapter 4, a geometric modification to the conventional uniform circular antenna array was proposed. The modification consisted in placing one of the antenna elements at the centre of the array. It was shown that, given the appropriate phase shift to this central element, the antenna directivity was increased while the half-power beamwidth angle was reduced. This resulted in better capability of transmitting the maximum power towards the desired direction and of avoiding unwanted signals. It was also observed that the sidelobe levels of the radiation pattern were lower than those of the conventional circular antenna array.

Throughout this thesis, it has been noted that in order for an antenna system to be "smart", a sophisticated signal processing control is needed. Due to the impressive development of computers, the application of optimization algorithms to antenna problems has become feasible. Bio-inspired algorithms like Genetic Algorithm and Particle Swarm Optimization are being applied to adapt the response of an antenna array in order to reject interference. These techniques have proven to be successful in the antenna array arena, from the physical design of antenna elements to beamforming on large antenna arrays. The application of these optimization algorithms to control the characteristics of the antenna pattern have been employed in Chapters 5 and 7. Chapter 5 focuses on the application of the Particle Swarm Optimization method to obtain a set of phase shift weights to be used in an adaptive antenna array. These weights allow the system to maximize the power of the main beam in a desired direction while reducing interference by

placing nulls toward jammers. The results obtained by the PSO algorithms were compared with those produced by the conventional Genetic Algorithm. It was observed that the PSO achieves better and more consistent radiation patterns. In addition, the total number of fitness function evaluations was lower for the PSO. This suggests an advantage in terms of computational cost as the function evaluation represents a high percentage of the total performance.

Chapter 6 proposed a similar use of Particle Swarm Optimization for beamforming but the criteria for selecting weights was changed. The fitness function used the signal-to-interference-plus-noise ratio. Also in this chapter, the antenna configuration was changed to a uniform rectangular array. The weights were optimized by maximizing the SINR which causes an increase in power toward the desired direction and a decrease in the direction of interferers. A standard Genetic Algorithm was also programmed to use this fitness function and the results were compared to those from the PSO.

In Chapter 7, the importance of choosing the right initial optimization parameters was stressed. A technique called meta-optimization was introduced which consists of employing a second optimization layer to find the best behavioural parameters for a given algorithm. This technique, also called super-optimization or meta-evolution had not previously been used for antenna optimization problems so its feasibility had to be proven. To accomplish this, the meta-landscapes for different optimization algorithms solving an antenna problem were obtained. A series of experiments was carried out in order to compare the algorithm performance when using standard, hand-tuned and meta-optimized parameters. In the case of the PSO algorithm, the change in velocity and position was studied in order to observe the behaviour of the particles. It was found that the particles converged faster when using the meta-optimized parameters compared with the velocities achieved with the standard and hand-tuned parameters. Furthermore, an antenna synthesis problem was also tackled using meta-optimization. Several meta-optimizers were used to find the best parameters that configured the PSO algorithm so as to find the optimum distances between elements in a problem presented previously by other researchers.

8.3 Conclusions

This thesis has investigated the geometrical characteristics of uniform circular antenna arrays and how they affect the overall radiation pattern. This was carried out by analysing different antenna configurations. A geometric modification to the conventional uniform circular antenna array was proposed in which one of the antenna elements is placed at the centre of the array. In Chapter 4, measurements of directivity, half-power beamwidth and sidelobe levels were performed and it was found that the radiation pattern can be modified in such a way that the directivity is increased while the half-power beamwidth angle is reduced. The result is a better capability of transmitting in the desired direction and avoiding unwanted signals. In the same chapter it was also observed that the sidelobe levels of the radiation pattern were lower than the ones of the conventional circular antenna array which also helps to avoid interference. It was concluded that, to obtain the best trade-off, the circular array should be conformed by 6 antenna elements, which is the configuration that shows better directivity and reduced half-power beamwidth.

Furthermore, this work focused on the feasibility of using bio-inspired algorithms to control adaptive antenna arrays. Algorithms like Particle Swarm Optimization and Genetic Algorithm were used to obtain the optimal antenna radiation pattern for a given problem. A comparison between these two algorithms was carried out and the results show that the PSO achieves better and more consistent radiation patterns than those of the GA. It was also observed that the total number of fitness function evaluations is lower for the PSO, which suggests an advantage in terms of performance as the function evaluation tends to have higher computational cost. Furthermore, the signal-to-interference-plus-noise ratio was used in these algorithms as part of the fitness function. The Particle Swarm Optimization method was used to generate a set of array weights to configure a planar rectangular array. These weights were optimized in order to maximize the power towards a desired direction whilst minimizing in the direction of interferers. A standard Genetic Algorithm was also studied and the results show that the PSO performs better in terms of the total number of fitness function evaluations. This suggests an advantage in performance as the fitness

function tends to have a high computational cost. It was also observed that the PSO obtains on average lower sidelobe levels which are desirable to avoid interference. In addition, the gain levels in the direction of nulls were computed and it was found that the PSO produces lower values than those of the GA.

Finally, this thesis investigated strategies that enhanced the effectiveness of the optimization algorithm. It was found that the selection of the right initial optimization parameters was of importance in order to obtain better results. A technique called meta-optimization was used to find the best behavioural parameters for a given antenna problem. As this technique has never been applied to antenna optimization, a series of experiments was carried out to study the feasibility of this approach. Meta-landscapes for different optimization schemes were presented and it was concluded that the problem of finding better parameters was relatively simple when using non-complex algorithms like PS and LUS. Experiments were carried out in which an antenna problem was solved by using standard, hand-tuned and meta-optimized parameters. The results showed better progress in terms of the number of iterations as well as consistency in the values of fitness functions. A classical problem in antenna synthesis was also studied, namely the use of Particle Swarm Optimization for linear array synthesis with minimum sidelobe and null control. A set of optimization parameters was obtained and used to configure the PSO algorithm to calculate the distance between elements for the mentioned linear array. It was found that the meta-optimized PSO outperformed the results found in the literature. The present thesis provides a better understanding of the configuration parameters used by different optimization algorithms when solving adaptive antenna array problems.

8.4 Future Work

At the conclusion of this thesis, there are issues that still need further investigation:

- This work focused on the maximization of the signal-to-interference-plus-noise ratio as the criteria used by the optimization algorithms. However, other avenues can be investigated in order to expand into a real world

application where additional information can be obtained. Transmission power, Bit-Error-Rate and Quality of Service amongst others are parameters that can be used in addition to the SINR. Furthermore, it is necessary to consider different scenarios such as multi-path interference and other background noise models in order to approximate the experiments to real situations.

- The algorithms used in this thesis were the "standard" ones, in other words, the basic versions. This leaves a lot of possible avenues open to developments in terms of using the more advanced and optimized versions. In particular, there are several versions of the PSO algorithm that have been adapted to solve antenna problems, for example the improved PSO presented in [107] and the Enhanced PSO (EPSO) described in [108] among others [39, 109]. In future work, these versions can be meta-optimized and their performance compared with each other.

- For performance reasons, the simulations carried out in this thesis for meta-optimization were limited to linear arrays. With the appropriate computational power, larger configurations of antennas could be tested.

Acronyms

AF	Array Factor
BA	Bees Algorithm
DE	Differential Evolution
DMTL	Distributed MEMS Transmission Line
DOA	Direction-of-Arrival
DSP	Digital Signal Processing
EPSO	Extended Particle Swarm Optimization
FPGA	Field Programmable Gate Arrays
GA	Genetic Algorithm
HC	Hill Climb
HPBW	Half-Power Beamwidth
LMS	Least Mean Squares
LUS	Local Unimodal Sampling
MEMS	Micro-Electro-Mechanical Systems
MIMO	Multiple-Input-Multiple-Output
MSE	Mean-Square Error
PS	Pattern Search

PSO Particle Swarm Optimization

QPM Quadratic Programming Method

RLS Recursive Least Squares

SA Simulated Annealing

SDMA Space Division Multiple Access

SINR Signal-to-Interference-plus-Noise Ratio

SLC Side-Lobe Cancellation

SLL Side-Lobe Level

SNOI Signals-Not-of-Interest

SOI Signals-of-Interest

UCA Uniform Circular Array

ULA Uniform Linear Array

URA Uniform Rectangular Array

List of Figures

List of Tables

Bibliography

[1] T. Heath, R. Kerr, and G. Hopkins, "Two-dimensional, nonlinear oscillator array antenna," in *Aerospace Conference, 2005 IEEE*, march 2005, pp. 1104 –1115.

[2] N. Jin and Y. Rahmat-Samii, "Parallel PSO/FDTD algorithm for the optimization of patch antennas and ebg structures," in *Wireless Communications and Applied Computational Electromagnetics, 2005. IEEE/ACES International Conference on*, april 2005, pp. 582 – 585.

[3] A. Mahanfar, S. Bila, M. Aubourg, and S. Verdeyme, "Design of planar microwave filters using a simple fdtd model and particle swarm optimization," in *Antennas and Propagation Society International Symposium, 2005 IEEE*, vol. 2B, july 2005, pp. 259 – 262 vol. 2B.

[4] P. Ioannides and C. Balanis, "Uniform circular and rectangular arrays for adaptive beamforming applications," *Antennas and Wireless Propagation Letters, IEEE*, vol. 4, pp. 351 – 354, 2005.

[5] ——, "Uniform circular arrays for smart antennas," *Antennas and Propagation Magazine, IEEE*, vol. 47, no. 4, pp. 192 – 206, aug. 2005.

[6] V. Zuniga, N. Haridas, A. T. Erdogan, and T. Arslan, "Effect of a central antenna element on the directivity, half-power beamwidth and sidelobe level of circular antenna arrays," *NASA/ESA Conference on Adaptive Hardware and Systems (AHS-2009)*, pp. 252–256, San Francisco, California, USA, July 29 - August 1 2009.

[7] R. Haupt, "Adaptive antenna arrays using a genetic algorithm," in *Adaptive and Learning Systems, 2006 IEEE Mountain Workshop on*, july 2006, pp. 249 –254.

[8] H. Wang, D.-G. Fang, and Y. Chow, "Grating lobe reduction in a phased array of limited scanning," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 6, pp. 1581 –1586, june 2008.

[9] K. Guney and M. Onay, "Amplitude-only pattern nulling of linear antenna arrays with the use of bees algorithm," *Progress In Electromagnetics Research*, vol. 70, pp. 21–36, 2007.

[10] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*. Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.

[11] D. Gies and Y. Rahmat-Samii, "Particle swarm optimization for reconfigurable phase-differentiated array design," *Letters on microwave optimization technologies*, vol. 38, pp. 168–175, August 2003.

[12] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection." *Information Processing Letters*, pp. 317–325, 2003.

[13] A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124 –141, jul 1999.

[14] S. Smit and A. Eiben, "Comparing parameter tuning methods for evolutionary algorithms," in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, may 2009, pp. 399 –406.

[15] C. Khodier, M.and Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 8, no. 53, pp. 2674–2679, August 2005.

[16] F. Gross, *Smart antennas for wireless communications*, 1st ed. McGraw-Hill, 2005.

[17] L. Van Atta, "Electromagnetic reflection," in *U.S. Patent 2908002*, October 6, 1959.

[18] R. Walden, "Performance trends for analog to digital converters," *Communications Magazine, IEEE*, vol. 37, no. 2, pp. 96 – 101, feb 1999.

[19] N. Haridas, R. Zhang, A. El-Rayis, A. Erdogan, T. Arslan, A. Bunting, and A. Walton, "Multiband micro antenna on silicon substrate," in *Aerospace conference, 2009 IEEE*, march 2009, pp. 1 –7.

[20] W. L. Stutzman and G. A. Thiele, *Antenna Theory and Design*, 2nd ed. Wiley, 1997.

[21] IEEE, IEEE standard definitions of terms for antennas, 22 Jun 1983, e-ISBN: 0-7381-4343-X.

[22] S. J. Orfanidis, *Electromagnetic Waves and Antennas*, 1st ed. Rutgers University, 2008.

[23] C. A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed. Arizona State University, 2005.

[24] ——, "Smart antennas for future reconfigurable wireless communication networks," Arizona State University, Tech. Rep., April 2000.

[25] J. Litva and T. K.-Y. Lo, *Digital beamforming in wireless communications*. Artech House, 1996.

[26] R. T. Compton, *Adaptive antennas: Concepts and performance*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[27] D. S. Weile and E. Michielssen, "The control of adaptive antenna arrays with genetic algorithms using dominance and diploidy," *IEEE transactions on antennas and propagation*, vol. 49, no. 10, pp. 1424–1433, October 2001.

[28] W. P. Keizer, "Fast low-sidelobe synthesis for large planar array antennas utilizing successive fast fourier transform of the array factor," *Antennas and Propagation, IEEE Transactions on*, vol. 55, no. 3, pp. 715 – 730, March 2007.

[29] Y. Rahmat-Samii and C. G. Christodoulou, "Guest editorial for the special issue on synthesis and optimization techniques in electromagnetics and antenna system design," *Antennas and Propagation, IEEE Transactions on*, vol. 55, no. 3, pp. 518 –522, march 2007.

[30] J. H. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press*, 1975.

[31] D. E. Goldberg, "Genetic algorithms in search optimization and machine learning," *Addison-Wesley*, 1989.

[32] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 13, 1983.

[33] M. Vecchi and S. Kirkpatrick, "Global wiring by simulated annealing," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 215 – 222, october 1983.

[34] S. Luke, C. Domeniconi, K. D. Jong, J. Grefenstette, C. Vo, J. Harrison, K. Sullivan, B. Hrolenok, B. Langdon, R. P. Wiegand, and et al., *Essentials of Metaheuristics A Set of Undergraduate Lecture Notes by Sean Luke.* Department of Computer Science, George Mason University, 2011.

[35] R. Storn and K. Price, "Differential evolution - a simple and effcient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[36] R. Storn, "On the usage of differential evolution for function optimization," *Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pp. 519–523, June 1996.

[37] C. Darwin, "On the origin of species (by means of natural selection, or the preservation of favoured races in the struggle for life)," 1859.

[38] R. Haupt, "Phase-only adaptive nulling with a genetic algorithm," *Antennas and Propagation, IEEE Transactions on*, vol. 45, no. 6, pp. 1009 –1015, jun 1997.

[39] M. Donelli, R. Azaro, F. G. B. D. Natale, and A. Massa, "An inovative computational approach based on a particle swarm strategy for adaptive phased-arrays control," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 3, pp. 888–897, March 2006.

[40] H. Bersini, M. Dorigo, S. Langerman, G. Seront, and L. M. Gambardella, "Results of the first international contest on evolutionary optimisation (1st ICEO)," in *International Conference on Evolutionary Computation*, 1996, pp. 611–615.

[41] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing Series. Springer, 2005.

[42] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[43] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[44] T. Krink and M. LÃ¸vbjerg, "The lifecycle model: Combining particle swarm optimisation, genetic algorithms and hillclimbers," in *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, 2002, pp. 621–630.

[45] S. Chalup and F. Maire, "A study on hill climbing algorithms for neural network training," in *Proceedings of the Congress on Evolutionary Computation (CEC'99), Washington D.C*, 1999, pp. 2014–2021.

[46] R. Hooke and T. Jeeves, ""Direct Search" solution of numerical and statistical problems," *Journal of the Association for Computing Machinery (ACM)*, vol. 8, no. 2, pp. 212 – 229, 1961.

[47] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, pp. 1–25, 1993.

[48] W. C. Davidon, "Variable metric method for minimization," *Journal on Optimization (SIAM)*, vol. 1, no. 1, pp. 1–17, 1991.

[49] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American Mathematical Society*, vol. 4, no. 3, pp. 502–506, 1953.

[50] M. E. H. Pedersen, "Tuning & simplifying heuristical optimization," Ph.D. dissertation, University of Southampton, 2010.

[51] A. C. M.E.H. Pedersen, "Local unimodal sampling," *Hvass Laboratories Technical Report no. HL0801*, 2008.

[52] R. Luus and T. Jaakola, "Optimization by direct search and systematic reduction of the size of search region," *American Institute of Chemical Engineers Journal (AIChE)*, vol. 19, no. 4, pp. 760–766, 1973.

[53] R. C. Johnson, *Antenna Engineering Handbook*, 3rd ed. McGraw-Hill, 1993.

[54] A. Ben and G. Mohammad, *Adaptive Array Systems: Fundamentals and Applications*, 1st ed. John Wiley & Sons, Ltd., 2005.

[55] Y. C. Chung and R. Haupt, "Optimum amplitude and phase control for an adaptive linear array using a genetic algorithm," in *Antennas and Propagation Society International Symposium, 1999. IEEE*, vol. 2, aug 1999, pp. 1424 –1427 vol.2.

[56] H. Zhao, Z. Xie, H. Wang, and J. Jin, "Beam shaping for satellite phased array antenna using dual coding genetic algorithm," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, sep 2009, pp. 1 –4.

[57] G. Mahanti, A. Chakraborty, and S. Das, "Floating-point genetic algorithm for design of a reconfigurable antenna arrays by phase-only control," in *Microwave Conference Proceedings, 2005. APMC 2005. Asia-Pacific Conference Proceedings*, vol. 5, dec. 2005, p. 3 pp.

[58] P. J. Bevelacqua, "Antenna arrays: Performance limits and geometry optimization," Ph.D. dissertation, Arizona State University, May 2008.

[59] B. D. Satish, "Comparative analysis of antenna beam patterns of the 6-by-1 linear array and the 6-by-2 rectangular array," *International conference on computer Communication and Control*, pp. 152–161, 23-25th Nov 2006.

[60] J. Foutz, A. Spanias, S. Bellofiore, and C. Balanis, "Adaptive eigenprojection beamforming algorithms for 1d and 2d antenna arrays," *Antennas and Wireless Propagation Letters, IEEE*, vol. 2, no. 1, pp. 62 –65, 2003.

[61] M. M. Sohul, "Impact of antenna array geometry on the capacity of mimo communication system," in *Electrical and Computer Engineering, 2006. ICECE '06. International Conference on*, dec. 2006, pp. 80 –83.

[62] A. Abouda, H. El-Sallabi, and S. Haggman, "Impact of antenna array geometry on mimo channel eigenvalues," in *Personal, Indoor and Mobile Radio Communications, 2005. PIMRC 2005. IEEE 16th International Symposium on*, vol. 1, sept. 2005, pp. 568 –572.

[63] P. Bevelacqua and C. Balanis, "Optimizing antenna array geometry for interference suppression," *Antennas and Propagation, IEEE Transactions on*, vol. 55, no. 3, pp. 637 –641, march 2007.

[64] R. Haupt, "Optimized element spacing for low sidelobe concentric ring arrays," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 1, pp. 266 –268, jan. 2008.

[65] M. Dessouky, H. Sharshar, and Y. Albagory, "Efficient sidelobe reduction technique for small-sized concentric circular arrays," *Progress In Electromagnetics Research*, vol. PIER 65, pp. 187–200, 2006.

[66] F. P. and P. Darwood, "Beamforming for circular and semicircular array antennas for low-cost wireless lan data communications systems," in *IEE Proc. Microwaves, Antennas and Propagation*, vol. 145, April 1998, pp. 153–158.

[67] B. L. and C. Comsa, "Analysis of circular arrays as smart antennas for cellular networks," in *Proc. IEEE Int. Symp. Signals, Circuits andSystems*, vol. 2, July 2003, pp. 525–528.

[68] K. Gyoda and T. Ohira, "Design of electronically steerable passive array radiator (ESPAR) antennas," *Antennas and Propagation Society International Symposium*, vol. 2, pp. 922–925, July 2000.

[69] R. J. Barton, P. J. Collings, P. E. Crittenden, M. J. Havrillas, and A. J. Terzuoli, "A compact passive broadband hexagonal spiral antenna array," *International Geoscience and Remote Sensing Symposium*, pp. 593–595, July 2007.

[70] C. Microstripes, http://www.cst.com, [Accessed April, 2011].

[71] T. Arslan, N. Haridas, E. Yang, A. Erdogan, N. Barton, A. Walton, J. Thompson, A. Stoica, T. Vladimirova, K. McDonald-Maier, and W. Howells, "ESPACENET: A framework of evolvable and reconfigurable sensor networks for aerospace based monitoring and diagnostics," in *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, june 2006, pp. 323 –329.

[72] N. Haridas, A. T. Erdogan, T. Arslan, A. J. Walton, S. Smith, T. Stevenson, C. Dunare, A. Gundlach, J. Terry, P. Argyrakis, K. Tierney, A. Ross, and T. O'Hara, "Reconfigurable mems antennas," in *Proc. NASA/ESA Conf. Adaptive Hardware and Systems AHS '08*, 2008, pp. 147–154.

[73] N. Haridas, A. El-Rayis, A. T. Erdogan, and T. Arslan, "Multi-frequency antenna design for space-based reconfigurable satellite sensor node," in *Proc. Second NASA/ESA Conf. Adaptive Hardware and Systems AHS 2007*, 2007, pp. 14–19.

[74] N. Haridas, A. T. Erdogan, T. Arslan, and M. Begbie, "Adaptive micro-antenna on silicon substrate," in *Proc. First NASA/ESA Conf. Adaptive Hardware and Systems AHS 2006*, 2006, pp. 43–50.

[75] C.-H. Hsu, "Optimizing beam pattern of adaptive linear phase array antenna using local genetic algorithm," in *Antennas and Propagation Society International Symposium, 2005 IEEE*, vol. 1B, 2005, pp. 315 –318 vol. 1B.

[76] R. C. Eberhart and Y. Shi, "Evolving artificial neural networks," in *Proc. 1998 Int. Conf. Neural Networks and Brain*, Beijing, 1998.

[77] S. M. Mikki and A. A. Kishk, "Quantum particle swarm optimization for electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 54, no. 10, pp. 2764–2775, October 2006.

[78] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 2, no. 52, pp. 397–407, February 2004.

[79] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna," in *Proc. IEEE Int. Symp. Antennas Propagation*, ser. vol. 1, San Antonio, TX, 2002, pp. 314–317.

[80] J. Kennedy and W. M. Spears, "Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on multi modal problem generator," in *Proc. IEEE Int. Conf. Evolutionary Computation*, 1998.

[81] G. Kokai, T. Christ, and H. Frhauf, "Using hardware-based particle swarm method for dynamic optimization of adaptive array antennas," in *Adaptive Hardware and Systems, 2006. AHS 2006. First NASA/ESA Conference on*, june 2006, pp. 51 –58.

[82] W. Li, Y. Hei, X. Shi, S. Liu, and Z. Lv, "An extended particle swarm optimization algorithm for pattern synthesis of conformal phased arrays,"

International Journal of RF and Microwave Computer-Aided Engineering, vol. 20, no. 2, pp. 190–199, March 2010.

[83] M. Shihab, Y. Najjar, N. Dib, and M. Khodier, "Design of non-uniform circular antenna arrays using particle swarm optimization," *Journal of electrical engineering*, vol. 59, no. 4, pp. 216–220, 2008.

[84] C.-H. Hsu, W.-J. Shyr, and C.-H. Chen, "Adaptive pattern nulling design of linear array antenna by phase-only perturbations using memetic algorithms," *Innovative Computing, Information and Control, 2006. ICICIC*, vol. 3, pp. 308–311, Aug 2006.

[85] M. Clerc, http://clerc.maurice.free.fr/pso/, [Accessed: April 2011].

[86] R. C. Eberhart and Y. Shi, "Particle swarm optimization: Developments, aplications and resources," in *Proc. 2001 Congress on Evolutionary Computation*, ser. vol. 1, 2001, pp. 81–86.

[87] S. Applebaum, "Adaptive arrays," *Antennas and Propagation, IEEE Transactions on*, vol. 24, no. 5, pp. 585 – 598, Sep. 1976.

[88] P. Howells, "Intermediate frequency sidelobe canceller," in *U.S. Patent 3202990*, August 24, 1965.

[89] ——, "Explorations in fixed and adaptive resolution at GE and SURC," *Antennas and Propagation, IEEE Transactions on*, vol. 24, no. 5, pp. 575 – 584, sep 1976.

[90] B. Widrow, P. Mantey, L. Griffiths, and B. Goode, "Adaptive antenna systems," *Proceedings of the IEEE*, vol. 55, no. 12, pp. 2143 – 2159, dec. 1967.

[91] M. Benedetti, R. Azaro, D. Franceschini, and A. Massa, "PSO-based real-time control of planar uniform circular arrays," *IEEE Transactions on Antennas and Propagation*, vol. 5, pp. 545–548, 2006.

[92] R. Monzingo and T. Miller, *Introduction to Adaptive Arrays*. Wiley, New York, 1980.

[93] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58 –73, feb 2002.

[94] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," *Lecture Notes in Computer Science*, vol. 1447, pp. 591–600, 1998.

[95] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, 2000, pp. 84 –88.

[96] R. Mercer and J. Sampson., "Adaptive search using a reproductive metaplan," in *The International Journal of Systems and Cybernetics*, ser. 7, 1978, pp. 215 – 228.

[97] J. Grefenstette, "Optimization of control parameters for genetic algorithms," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 16, no. 1, pp. 122 –128, jan. 1986.

[98] R. E. Bellman, *Dynamic programming*. Princeton University Press, 1957.

[99] M. E. H. Pedersen. (2011) Swarmops. numerical & heuristic optimization. [Online]. Available: http://www.hvass-labs.org/projects/swarmops/

[100] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," *Proceedings of MENDEL 2002, 8th International Mendel Conference on Soft Computing*, pp. 62–67, 2002.

[101] A. Carlisle and G. Dozier, "An of-the-shelf PSO," *Proceedings of the Particle Swarm Optimization Workshop*, pp. 1–6, 2001.

[102] M. M. Khodier and M. Al-Aqeel, "Linear and circular array optimization: a study using particle swarm intelligence," *Progress In Electromagnetics Research B*, vol. 15, pp. 347–373, 2009.

[103] Y. Rahmat-Samii and N. Jin, "Particle swarm optimization (pso) in engineering electromagnetics: A nature-inspired evolutionary algorithm," in

Electromagnetics in Advanced Applications, 2007. ICEAA 2007. International Conference on, sep 2007, pp. 177 –182.

[104] N. Petrella, M. Khodier, M. Antonini, M. Ruggieri, S. Barbin, and C. Christodoulou, "Planar array synthesis with minimum sidelobe level and null control using particle swarm optimization," in *Microwaves, Radar Wireless Communications, 2006. MIKON 2006. International Conference on*, May 2006, pp. 1087 –1090.

[105] D. Mandal, A. Bhattacharjee, and S. Ghoshal, "Comparative optimal designs of non-uniformly excited concentric circular antenna array using evolutionary optimization techniques," in *Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on*, Dec. 2009, pp. 619 –624.

[106] ——, "A novel particle swarm optimization based optimal design of three-ring concentric circular antenna array," in *Advances in Computing, Control, Telecommunication Technologies, 2009. ACT '09. International Conference on*, Dec. 2009, pp. 385 –389.

[107] D. Mandal, S. Ghoshal, and A. Bhattacharjee, "Improved swarm intelligence based optimal design of concentric circular antenna array," in *Applied Electromagnetics Conference (AEMC), 2009*, Dec. 2009, pp. 1 –4.

[108] M. Mangoud and H. Elragal, "Wide null beamforming using enhanced particle swarm optimization," in *Communications (MICC), 2009 IEEE 9th Malaysia International Conference on*, dec 2009, pp. 159 –162.

[109] M. Benedetti, R. Azaro, and A. Massa, "Memory enhanced pso-based optimization approach for smart antennas control in complex interference scenarios," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 7, pp. 1939 –1947, july 2008.